A bi-objective model for water allocation and scheme water scheduling

Johannes Jacobus Barnard

Thesis presented in partial fulfilment of the requirements for the degree of
Master of (Industrial) Engineering
in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr Daniel Lötter

April 2019

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2019

Abstract

"Water is life" — this common expression is often referred to as a cliche, but Earth's inhabitants can truly bear witness to the accuracy of this statement. As a result of global warming, the El Ninõ phenomenon, a growing population and expanding economies, water has globally become a cherished commodity. In the Western Cape, South Africa, a devastating drought experienced over the two-year period 2017–2018 has propelled the innovation of more effective and efficient water management strategies to the forefront, especially in the farming sector, where farmers are currently compelled to produce agricultural crops with less water. An irrigation and scheme water supply schedule may, for example, be employed with the aim of proposing how crops should be irrigated during their various growth stages, if natural water supply is insufficient, and how additional scheme water supply should be scheduled to enhance efficient water use. An open-air irrigation reservoir typically serves as a water storage facility for the purpose of irrigating agricultural crops. Evaporation (the process of transforming water vapour into the atmosphere) from such a reservoir water surface may, however, result in a reduced reservoir capacity of up to 20%.

In this thesis, two novel mathematical models are proposed which form the basis of a decision support system for farmers aimed at providing beneficial agricultural crop irrigation strategies. The first is a single-objective optimisation model which proposes an irrigation schedule in conjunction with a scheme water supply schedule in which the goal is to maximise the total profit obtainable from crop yield. This maximisation process is subject to a user-specified reservoir water capacity that should be left over in an open-air reservoir at the end of a specified scheduling horizon. If possible, additional water resources may be obtained from scheme water supply in order to aid with the irrigation of crops. These additional water resources, however, usually come at a cost, which is also included in the total profit calculation.

The second model is a bi-objective optimisation model which aims to maximise the total profit from crop yield while simultaneously maximising the reservoir water contents at the end of the last scheduling period. When plotted in objective space, the solutions to the model form a Pareto front that is presented as the basis of decision support to the decision maker (farmer), providing him with an overview of numerous implementable irrigation and scheme water supply schedules for a variety of end-period reservoir water levels.

Both the single-objective and bi-objective optimisation models are validated in three ways, namely by face validation, by random benchmark validation and by consulting an expert in the field of crop irrigation and farming. Embedded in the decision support system, these models enable the decision maker to develop a course of action in terms of crop irrigation for a tailored farming scenario.

Uittreksel

"Water is lewe" — daar word dikwels na hierdie algemene uitdrukking as 'n cliche verwys, maar die Aarde se populasie kan werklik van die waarheid van hierdie stelling getuig. As gevolg van aardverwarming, die El Ninõ-fenomeen, 'n toenemende bevolking en groeiende ekonomieë, het water wêreldwyd 'n gekoesterde hulpbron geword. In die Wes-Kaap, Suid-Afrika het 'n ernstige droogte oor die tweejaar-periode 2017–2018 dringende innovasie van doeltreffender waterbestuurstrategieë genoop, veral in die boerderysektor, waar boere tans verplig is om landbougewasse met minder water te kweek. 'n Besproeiings- en skemavoorsieningskedule kan byvoorbeeld aangewend word om te bepaal hoe gewasse tydens hul verskillende groeistadia besproei moet word indien natuurlike water beskikbaarheid onvoldoende is, en hoe addisionele skemavoorsiening geskeduleer behoort te word om doeltreffende watergebruik te verhoog. 'n Opelugbesproeiingsreservoir dien gewoonlik as 'n waterbergingsfasiliteit vir die besproeiing van landbougewasse. Verdamping (die omskakelingsproses van waterdamp na die atmosfeer) vanaf só 'n reservoirwateroppervlak kan egter na 'n verminderde reservoir-kapasiteit van tot 20% lei.

Twee nuwe wiskundige modelle word in hierdie tesis voorgestel wat as basis vir 'n besluitsteunstelsel vir boere kan dien en wat daarop gemik is om voordelige besproeiingstrategieë vir landbougewasse te verskaf. Die eerste is 'n enkeldoelige optimeringsmodel wat 'n waterbesproeiingskedule tesame met 'n skema-watervoorsieningskedule voorstel, met die doel om die totale wins wat uit gewasopbrengs verkry kan word, te maksimeer. Hierdie maksimeringsproses is onderworpe aan 'n gebruikersgespesifiseerde reservoir-waterkapasiteit wat aan die einde van 'n gespesifiseerde skeduleringshorison in 'n opelugreservoir moet oorbly. Indien moontlik, kan bykomende waterhulpbronne uit skemavoorsiening vir die besproeiing van gewasse gebruik word. Hierdie addisionele waterhulpbronne is egter gewoonlik teen 'n koste beskikbaar, wat ook in die totale winsberekening ingesluit word.

Die tweede model is 'n twee-doelige optimeringsmodel wat daarop gemik is om die totale wins wat uit gewasopbrengs verkry kan word, te maksimeer, en terselfdertyd die waterinhoud van die reservoir aan die einde van die laaste skeduleringsperiode te maksimeer. Wanneer die oplossings van hierdie model in die doelfunksieruimte uitgestip word, word 'n Pareto-front verkry wat as die basis vir besluitsteun aan die besluitnemer (boer) voorgelê kan word, en aan hom 'n oorsig bied oor die verskeidenheid van implementeerbare besproeiings- en skemavoorsieningskedules vir verskeie oorblywende reservoir-watervlakke aan die einde van die laaste skeduleringsperiode.

Beide die enkel-doelige en die twee-doelige optimeringsmodelle word op drie maniere gevalideer, naamlik deur sigvalidering, deur lukrake toetsvalidering en deur 'n deskundige op die gebied van gewasbesproeiing en boerdery te raadpleeg. Hierdie modelle word in die besluitsteunstelsel geïnkorporeer en stel die besluitnemer sodoende in staat om 'n plan van aksie in terme van gewasbesproeiing vir 'n gespesifiseerde boerdery-scenario te ontwikkel.

Acknowledgements

The author wishes to acknowledge the following people and institutions for their various contributions towards the completion of this work:

- To my study leader, Dr Daniel Lötter, for his support, encouragement and guidance during the course of this thesis. I truly appreciate his dedication, his time and his contribution towards the quality and originality of the work presented. His contributions stretches far beyond the tireless editing and improvements to this document. The imparting knowledge, thoughts and moments of laughter that was shared on a number of social occasions will be cherished.
- To the meticulous Prof Jan van Vuuren for his shared passion in the field of Operations Research — this is a true inspiration. I truly appreciate his willingness to help as well as the imparting knowledge that was shared on other important aspects in life.
- To my fellow SUnORE lab members whom participated alongside me in this thesis. Thank you for the laughter, patient assistance and memories that was shared during and after office hours. It was a privilege to share two years of my life with you.
- To my parents for providing me with the opportunity to pursue a masters degree in Engineering, and supporting me in a number of ways during the course of this thesis.

Table of Contents

List of Reserved Symbols

The following symbols are reserved for specific use in the respective contexts, unless otherwise specified in the localised section. Other symbols may be used throughout this thesis in an unreserved fashion.

Symbols in this thesis conform to the following font conventions:		
x_{max}, x, X	Symbol denoting a **parameter** or **variable**	(Upper or lower case)
$\boldsymbol{x}, \boldsymbol{X}$	Symbol denoting a **vector**	(Upper or lower case boldface)
$\mathcal{X}$	Symbol denoting a **set**	(Upper case calligraphy)

Multi-objective optimisation:

Symbol	Meaning
i_{max}	Number of computational iterations allowed before algorithmic termination
$maxepoch$	Number of epoch iterations allowed before reheating
α	The cooling parameter
$maxaccepts$	The initial acceptance ratio
$maxattempt$	The initial rejection ratio
$\mathcal{P}_S$	The set of Pareto optimal solutions
$\mathcal{A}$	The set of mutually nondominating solutions found thus far during the execution of the algorithm
$\mathcal{A}_{\text{vect}}$	The set of objective function vectors in the Pareto front found thus far during the execution of the algorithm
$\boldsymbol{x}$	The resulting solution vector that maximises the objective function after the execution of the algorithm

Two novel crop irrigation and scheme water supply models:

Symbol	Meaning
j	The index of different crops
t	The index of crop growth stages
p	The index of time periods
$\left(\frac{\boldsymbol{w}}{\boldsymbol{ET}}\right)_{\boldsymbol{jt}}$	The decision variable fragment denoting the amount of water (w) irrigated to crop j during growth stage t over the water requirement (ET) for crop j during growth stage t
$\boldsymbol{Z_p}$	The decision variable denoting a specific time period $p \in P$ scheduled for scheme water supply
$\boldsymbol{O_p}$	The decision variable denoting the amount of scheme water supply scheduled during time period $p \in P$

y_{jp}	A binary parameter taking the value 1 if crop $j \in J$ is grown during time period $p \in P$, or zero otherwise
Y_j^{avg}	The average yield (in tons per hectare) for crop $j \in J$
K_{jt}	The crop yield response factor (unitless) for crop $j \in J$ during growth stage $t \in T$
l_j	The number of hectares planted of crop $j \in J$
p_j	The selling price (in Rands per ton) for crop $j \in J$
v_j	The variable cost (in Rands per hectare) for crop $j \in J$
d_j	The fixed cost (in Rands) for crop $j \in J$
O_{price}	The cost for one m^3 of water
R_{cap}	The reservoir capacity
V_p	The end-period reservoir water volume for time period $p \in P$ where $p \neq P$
$V_{P_{end}}$	The end-period reservoir water volume for time period $p = P$
$U_{P_{end}}$	The user specified end-period reservoir volume that must be reached for time period $p = P$
$O_{p_{lower}}$	The minimum scheme water supply (in m^3 of water) for time period $p \in P$
$O_{p_{upper}}$	The maximum scheme water supply (in m^3 of water) for time period $p \in P$
O_{total}	The maximum accumulated scheme water supply (in m^3 of water) for time periods $p = \{1, 2, ..., P\}$
B_p	The amount of borehole flow (in m^3 of water) during time period $p \in P$
R_p	The amount of rainfall (in mm) during time period $p \in P$
W_{jt}	The water requirement for crop $j \in J$ during growth stage $t \in T$
e_p	The historical evaporation rate for time period $p \in P$
A_p	The reservoir water surface area for time period $p \in P$

List of Acronyms

AE: Average Error of bias

CRM: Coefficient of Residual Mass

CV: Coefficient of Variation

CWPF: Crop Water Production Function

DBMOSA: Dominance-Based Multi-Objective Simulated Annealing

DFD: Data Flow Diagram

DSS: Decision Support System

DSSAT: Decision Support System for Agrotechnology Transfer

EF: Modelling Efficiency

FAO: Food and Agriculture Organization

FAOSTAT: Food and Agriculture Organization STATistics

FNSA: Fast Nondominated Sorting Algorithm

FS: Full Season

GA: Genetic Algorithm

GUI: Graphical User Interface

ha: Hectares

***K*:** Potassium

LAI: Leaf Area Index

MAE: Mean Absolute Error

MOOP: Multi-Objective Optimisation Problem

MOSA: Multi-Objective Simulated Annealing

***N*:** Nitrogen

NCES: Natural Crop Estimates System

NP: Non-deterministic Polynomial

NSGA II: Nondominated Sorting Genetic Algorithm II

***P*:** Phosphorus

P: Polynomial

RMSE: Root Mean Square Error

SA: Simulated Annealing

SAPWAT: South African Program WATer

SOOP: Single-Objective Optimisation Problem

SWB: Soil Water Balance

UI: User Interface

VPD: Vapour Pressure Deficit

WPF: Water Production Function

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

Contents

1.1 Introduction

South Africa is generally described as a world in one country due to its astonishing biodiversity, wildlife and modern cities. With a surface area covering 1 219 602 km^2, South Africa stretches from 22°S to 35°S in latitude and 17°E to 33°E in longitude. The country shares boundaries with Namibia, Botswana, Zimbabwe, Mozambique, Swaziland and Lesotho, and has a coastline stretching over 3000 km. South Africa is, however, classified as a relatively dry country due to a below average annual rainfall of 464mm compared to the world average of 860mm annually [33]. Eventhough South Africa has a below average rainfall, it is still considered a very active country in terms of farming practices. The Land Bank, which is a specialist agriculture bank guided by the government mandate, contributed R40,9 billion on the country's *gross domestic product*[1] in order to ensure increasing job opportunities and financial growth in the farming industry. Furthermore, the gross income from all agricultural products for the year ending on 31 December 2016 was estimated at R259 620 million [33]. Farmers are, however, faced with an immense challenge since water resources in South Africa are limited, and their responsibility to provide for a growing nation is becoming larger as the availability of water resources are declining.

In the farming industry of South Africa, more than 50 000 smallholdings are responsible for approximately 1.3 million *hectares* (ha) of land under irrigation [133]. The land is irrigated using more than 50% of the surface water resources and a variety of other water schemes. In most cases, all large irrigation schemes are supplied from storage dams across the country, however, conveyance losses ensure that a great portion of water does not reach farmers. In the citrus industry, South Africa is considered the second largest exporter of citrus food. According

[1]The *gross domestic product* of a country denotes the financial value of all the finished products and services that are produced within a country within a specific time window. The gross domestic product also indicates economic wealth within a country, or in this case, within the agricultural sector [67].

to the UNEP Finance Initiative [133], 58 000 hectares account for 20 million trees of which oranges makes up 70% of the total yield. Citrus, and especially oranges, demand relatively high amounts of water because the quality of the Citrus products, as well as the quantity of the yield, are both functions of water quality and volume [133]. In the wine industry, on the other hand, vines account for 103 000 hectares of vineyards which are concentrated mostly in the Western and Northern Cape. Since the Western Cape relies on winter rainfall stretching from May to August, it is especially important for effective and efficient planning since table grapes require approximately 520–830 mm of water falling outside the rain season [133].

Water that is available to consumers in South Africa is constantly threatened by a number of factors. A growing population is one such factor which is responsible for an increase in water withdrawals and is due to the important role that water plays in the human body. Water acts as a transportation source which transports waste materials, carbohydrates and proteins within the body, and this is also responsible to keep the human body hydrated. It is estimated that the human body can only survive three days without water [6]. Furthermore, a growing population is the result of expanding businesses in emerging economies which lead to a growth in industrial and agricultural activities around the globe. Higher water withdrawals are required for production which result in constantly surpassing the ratio of water consumption over water availability [147].

Other factors that are responsible for reducing the availability of water include natural occurrences such as global warming and the El Niño phenomenon. Global warming is responsible for increasing temperatures globally and has a devastating effect on ecosystems, the ability to create food and the ability to develop socially [90]. Global warming is a result of carbon dioxide, greenhouse gasses and other air pollutants that gather in the atmosphere. These gasses are responsible for the reflection of radiation and sunlight from the earth's surface and back, and get trapped between the surface of the earth and the atmosphere. As a result of this, the earth's core temperatures are increasing. Furthermore, according to the CSIR, South Africa is one of the most vulnerable countries when it comes to global warming and that South Africa is heating up faster than the global average [90].

Higher temperatures have great consequences on the environment, which result in life forms to shift towards cooler environments in order to sustain livelihood, melting glaciers, rising sea levels, heat waves in conjunction with extreme droughts and weather conditions [80]. Moreover, the El Niño phenomenon is responsible for disrupting weather conditions and can either result in heavy rainfalls or drought conditions in various countries around the world [41]. An El Niño is a result of strong winds which push warm surface ocean water from South America towards Australia and Asia, and cold surface ocean water up to South America. This phenomenon is illustrated in Figure 1.1 and includes the temperatures of the ocean water in degrees *Celsius*. As a result, the warm surface water piles up on the east side of the pacific which causes ocean temperatures to rise, and contributes to excess energy that is released into the atmosphere causing temperatures to rise. Past studies on ocean warming indicated that nearly half of the industrial era increases in ocean warming, have occurred in the last twenty years since 1865. Ultimately, higher temperatures are responsible for increasing the rate of evaporation which may give rise to droughts.

The affect of a drought on a country could be devastating. The *Food and Agriculture Organization* (FAO) published a paper which examines the rehabilitation policies following a drought in order to aid recovery in crop and livestock production [129]. In this paper, the FAO specifically describe the devastating outcomes of a drought on crop yield and livestock. An immediate effect may be seen in the reduction of crop yield as rainfall is insufficient. Farmers may find it hard to maintain previous yield production standards since irrigating crops without sufficient rainfall

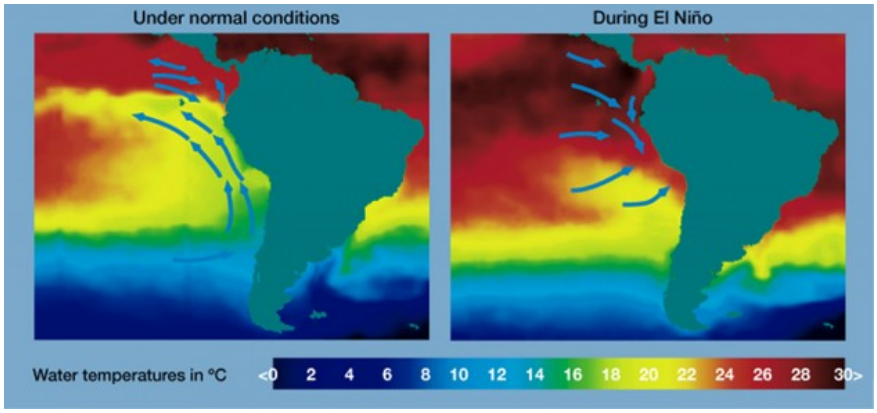


FIGURE 1.1: *The movement of warm surface water across South America due to strong winds [41].*

may result in water deficits, and thereby reducing gross profit achieved from crop yield when assuming that the farmer does not have access to additional water resources. In the case where water resources are available, extra water has to be bought in order to maintain the same production standards, otherwise crops must be irrigated with a limited water supply. Furthermore, livestock and assets may be sold to act as an economic buffer in times of hardship which may ultimately result in disinvesting these assets in order to survive.

In a country where the availability of water is threatened by global warming, the El Ninõ phenomenon, a growing population and expanding economies, efficient and effective planning of water resources should be considered to ensure the sustainability of water resources.

1.2 Informal problem description

Historically, regions within South Africa have been exposed to drought conditions many times which resulted in limited water supply and availability. The frequency of these conditions are increasing as a result of global warming patterns and the El Niño phenomenon. During these periods in which water resources are limited, farmers are forced to reduce irrigation to crops as a result of water restrictions imposed by government in a bid to reserve water for other uses such as human consumption. Reducing irrigation to crops affects the ability of crops to produce sufficient yield for a growing demand, and may also place a financial burden on farmers as a result of low crop yield. This, however, depends on the extent to which irrigation must be reduced.

The aim of thesis is to design a generic *decision support system* (DSS) that is capable of providing an irrigation strategy to farmers for irrigating crops that are currently grown while simultaneously scheduling scheme water supply[2] to promote effective and more efficient water management during times where water supply is limited. This approach relies on the interactions between economics, engineering and agriculture in a bid to aid farmers in their decision making process. When water supply is limited, farmers are faced with complexed decisions regarding the irrigation of their crops as well as which crops should be irrigated to achieve best financial results. A solution(s) to these complexed decisions is proposed by the DSS in order to support the decision making of farmers during such times.

[2]Scheme water supply refers to supplying water using a set of integrated facilities to a cluster of towns or nearby farmland [141].

The DSS proposed in this thesis should be able to take as input user-specified and location specific parameters in order to solve a realistic farm scenario using optimisation models. A single-objective and/or multi-objective optimisation model, depending on the preference of the decision maker, are employed to achieve this. In the single-objective model, the profit obtained from crop yield is maximised, while in the case of the multi-objective model, the profit obtained from crop yield and the end-period reservoir water capacity for a specified scheduling horizon are simultaneously maximised. The open-air reservoir located on the farm that is used to irrigate crops typically experience a number of activities that result in the depletion or augmentation of reservoir water volumes. These reservoir activities are also taken into consideration when proposing an irrigation schedule, a scheme water supply schedule, when estimating the reservoir water volume at the end of the scheduling horizon as well as estimating the total evaporation.

1.3 Project objectives

The following eight objectives are pursued in this thesis, namely:

I To *conduct* a comprehensive literature review on agricultural prerequisites related to:

(a) the basic concepts associated with evaporation, transpiration and estimating evaporation from water surface areas using historical A-pan evaporative data and a reservoir shape characteristic,

(b) the natural characteristics of crop growth stages and the affect thereof on evapotranspiration (the combination of evaporation and transpiration),

(c) the role that crop coefficients and yield response factors play in estimating the final crop yield using crop water production functions,

(d) statistically analysing the accuracy of crop water production functions when estimating final crop yield,

(e) existing crop production DSSs that incorporates crop water production functions or other methods of predicting final crop yield,

(f) a complete understanding of the working of CropWat 8.0 which plays a fundamental role in estimating crop coefficients and yield response factors, and

(g) soil moisture management systems which may be installed for estimating real time evaporation on farms.

II To *conduct* a comprehensive literature review on solution methodologies to solve crop water production functions in Objectives I(c) and (d) as an optimisation problem for single-objective and multi-objective models.

III To *formulate* a novel single-objective crop irrigation and scheme water supply model which is able to propose an irrigation strategy to irrigate crops while simultaneously provide a scheme water supply schedule which maximises the profit obtained from crop yield when water resources are limited. The model should incorporate all the notions in Objectives I(a)–(g) and is also subjected to a user-specified reservoir volume at the end of a specified time period.

IV To *adapt* the single-objective crop irrigation and scheme water supply model in Objective III to formulate a novel bi-objective crop irrigation and scheme water supply model that is able to maximise the total profit from crop yield while simultaneously maximising

the end-period reservoir water capacity at the end of the scheduling horizon. The results returned by the model is then an approximate front of optimal solutions from where the farmer may choose a solution for implementation purposes.

V To *validate* the two novel mathematical models formulated in Objectives III and IV using the results obtained from solving a small hypothetical scenario. A threefold validation procedure is followed. First, a face validation is done based on an initial generated solution from the two models, and followed by a random benchmark validation that is conducted on the results obtained from the small hypothetical scenario, and finally, consulting with an expert in the field of crop irrigation and farming to validate the model results, as well as the realism and authenticity of the proposed mathematical models in Objectives III and IV.

VI To *formulate* a realistic case study and to solve the two novel mathematical models in Objectives III and IV in the context of the case study using the solution methodologies in Objective II to illustrate the workability of these models in a real-world practical farming scenario. The case study should entail location specific parameters, the growing of location specific crops, a realistic hypothetical reservoir as well as the examination of two or more different crops that are grown.

VII To *design* a novel, generic *decision support system* which is capable of producing constructive feedback using generic graphs and tables for easier interpretation of the obtained results when solving the models in a specific farming scenario. The DSS should be able to incorporate both the mathematical models in Objectives III and IV to solve a realistic farm scenario as described in Objective VI that is developed within the DSS.

VIII To *propose* possible avenues for future work to be conducted on the research contained in this thesis.

1.4 Project scope

Due to multiple notions and concepts associated with the irrigation and growing of crops, the complexity involved in the proposed mathematical models can soon become large. This calls for employing a narrow down approach on a number of elements associated with irrigating crops. This thesis is, therefore, limited to the following assumptions:

Irrigation methods. There exist multiple irrigation methods which farmers may use to irrigate their crops. These irrigation methods include surface irrigation, sprinkler irrigation, drip irrigation and subsurface irrigation, of which the advantages and disadvantages differ from one another with respect to soil conditions and crop types [91]. These various types of irrigation methods are not considered in this thesis since the focus is placed on the amount of water irrigated to crops rather than the method to do so. The responsibility, therefore, lies with the farmer to ensure that the amount of water proposed by the models is irrigated to crops independent of the irrigation method used.

Time of irrigation. Farmers may irrigate crops at night in order to minimise evaporation losses and maintain sufficient soil moisture levels for longer time periods. Many farmers also make use of Ruraflex tariffs and chargers which involve lower electricity tariffs during time periods where demand is low. Although the practices may yield a reduction in costs, the time of irrigation is not be considered in this thesis since evapotranspiration is

inevitable during daytimes. For the purpose of this thesis, it is assumed that irrigation is applied during day time.

Water demand of plants. The mathematical models developed in this thesis incorporates crop water production functions in order to predict the actual yield gained from crops based on the evapotranspiration ratio, denoted as the ratio of the amount of water supplied over the amount of water required by crops. The amount of water irrigated to crops by farmers is, therefore, taken as an indication of the total water requirements by crops due to the risk of reduced crop yield when water requirements are not met.

Soil nutrients. Soil nutrients are regarded as a major factor contributing to plant growth and ultimately ensuring substantial crop yields. The most influential nutrients in the soil are known as NPK which stands for *nitrogen* (N), *phosphorus* (P) and *potassium* (K) [93]. Since this thesis focusses on proposing an irrigation schedule along with a scheme water supply schedule to irrigate crops effectively and efficiently when water supply is limited, it is assumed that the required nutrients to ensure full development of crops is maintained in the soil throughout the growing season of each crop.

Water resources. A number of additional methods may be implemented by farmers in order to obtain additional water resources such as for example the use of grey water for irrigation or the catchment of water from roof/store surfaces. In this thesis, only a number of sources are considered for obtained additional water resources and include borehole provision, pumping water from river basins, direct water from canals, as well as rainfall.

1.5 Thesis organisation

Excluding the introductory chapter, this thesis further comprises eight chapters in total. In Chapter 2, a comprehensive literature review is conducted on agricultural prerequisites which specifically aims to introduce readers to agricultural aspects related to growing and producing yield from crops through irrigation. As introduction to this chapter, a short summary is given on the history and practices that led to research development of irrigation in the United States. The notions of evaporation and transpiration are thoroughly discussed at the hand of crop water requirements and estimating evaporation from water surface areas due to the essential role these notions play in the irrigation of crops. A mathematical crop coefficient approach, containing multiple notions associated with estimating the final crop yield by means of water irrigated to crops is also briefly discussed. Each notion confined to this approach is thoroughly discussed as well as how to obtain specific parameter values when adopting such an approach. As part of estimating crop water requirements, which plays an important role in estimating the final crop yield, a concise description on soil moisture management systems in the farming industry are given. In the final section of this chapter the limitations of the adopted crop coefficient approach to estimate the crop final yield are critically discussed.

In Chapter 3, the notions related to multi-objective optimisation as part of the solution methodology adopted in this thesis are presented. Computational complexity as part of solving optimisation models using multi-objective solution approaches is defined and described as an introduction to the chapter. The convexity and non-convexity of multi-objective optimisation problems are also important aspects to consider when adopting specific solution approaches, and are thoroughly discussed in the next section. Moreover, solution dominance is inherent to multi-objective optimisation problems which demands an in-depth evaluation of this notion. This gives rise to the concept of Pareto optimality, strong dominance and weak Pareto optimality which are briefly

described. Related to the aforementioned notions, methods to determine the nondominated sets of solutions as part of a Pareto front are thoroughly discussed and pseudocode listings of these methods are also provided. Some solution methodologies demands multiple Pareto fronts, and may be obtained my employing the fast nondominated sorting algorithm which is also briefly described. The final section in this chapter describes one of the simplest methods to solve a multi-objective optimisation problem namely the weighted-sum of objectives method.

In Chapter 4, four solution methodologies that may be used to solve multi-objective optimisation problems are considered and described in detail to propose a solution method for a solving single-objective and/or multi-objective problem. As introduction to this chapter, exact solution approaches are thoroughly described and incorporates three popular exact solutions approaches. Next, approximate solution approaches known as heuristics when solving similar optimisation problems are also thoroughly described along with three well-known classes in this field, namely local search algorithms, constructive algorithms and iterative algorithms. Metaheuristics, on the other hand, are part of heuristic solution approaches but is different in nature. In the next section, two of the most popular metaheuristic solution approaches which may be employed to solve single-objective and multi-objective problems are briefly described. Pseudocode listings for each solution method are also provided.

Chapter 5 forms the heart of this thesis since it is here where two novel mathematical models which are able to propose an irrigation and scheme water supply schedule of high quality is proposed. The chapter opens with a number of assumptions based on the working dynamics of an open-air reservoir on which the mathematical models are formulated on. The proposed modelling framework considered in this thesis is described which proposes a careful examination of predicting rainfall measures. This results in two solution methodologies with respect to precipitation and the prediction thereof which are thoroughly described. The two novel optimisation models are introduced to the reader in the next section and are thoroughly described. Part of these models is obtaining an end-period reservoir water capacity which is obtained from periodic reservoir volumes. These calculations are briefly described in the context of a small hypothetical scenario in the closing section of this chapter.

In Chapter 6, the model implementation as implemented to solve a hypothetical scenario are presented to the reader. As introduction to this chapter, a small hypothetical instance are formulated and thoroughly described with the aim to present the working of the mathematical models described in Chapter 5. The mathematical models as implemented to solve the hypothetical instance are described at the hand of the adopted solution parameters and techniques. The model implementations and parameters specifically plays an important role in the model results. An in-depth parameter evaluation is conducted on these parameters in order to obtain optimal parameter values for best model results. The parameter evaluation does not entail a full factorial design as the computational time to solve such an experiment are dependent on the model size and complexity, which in this case are large and complex. The two novel mathematical models are then validated by means of the results obtained from solving the hypothetical instance using the optimal parameter values of which the results are then thoroughly discussed in the final section of the chapter.

Chapter 7 entails the introduction of a real-world case study that is solved by means of a preselected solution method from Chapter 4 and incorporating the mathematical models in Chapter 5. As introduction, the real world case study is presented which is then followed by a thorough evaluation on the model results when solving such a scenario considering that no rainfall took place. The chapter then closes with a re-evaluation on the updated model results when considering that rainfall did take place during some time periods.

In Chapter 8, a computerised DSS for solving a realistic farm scenario as described in Chapter 7 by incorporating the mathematical models in Chapter 5 is presented. In the first section, the basic notions of DSSs are briefly elucidated. In the second section, the *graphical user interface* of the DSS and the working of the DSS are thoroughly described, followed by a description on the general framework of the DSS. This chapter is then closed with a brief but concise discussion on the deployment and maintenance of the DSS.

This thesis close with a thesis summary in Chapter 9, followed by an appraisal on the thesis contributions. The chapter then closes with multiple suggestions for future research that may be conducted on the work presented in this thesis.

CHAPTER 2

Literature review: Agricultural prerequisites

Contents

In this chapter, the reader is introduced to a number of notions related to estimating crop yield when water irrigation is applied. These notions are especially important when strategic planning is done within farm environments. In §2.1, a brief introduction on irrigation practices and investigations in the United States of America, dating back to the 1500's, is provided. Next, in §2.2, the notions of evaporation and transpiration are thoroughly discussed since these notions are essential in estimating crop water requirements. This is then followed in §2.3 with a discussion on the solution approaches when estimating evaporation from water surface areas.

In §2.4, the notions related to computing the final yield from crops are briefly described. Within this section, crop water production functions lays the foundation for estimating crop yield when water is limited, and this is governed by crop yield response factors which are also discussed. This section concludes with a discussion on a number of crop production decision support systems from the literature. In §2.5, essential parts in soil moisture management systems are briefly discussed which is followed by a discussion on the limitations of adopting a crop yield response approach for estimating the final yield. Finally, the chapter closes in §2.7 with a brief review of the chapter contents.

2.1 A brief introduction of the history of irrigation and research

Irrigation practices and investigations in the field of water requirements for agricultural crops can be traced back as far as the 1500's [69]. Before the arrival of the Spaniards in the United States in 1598, irrigation has long been a prominent practice implemented by farmers in local communities, and as a result, some investigations regarding water requirements may be traced back hundreds of years ago. During the sixteenth century, the Spaniards also began to irrigate crops, and by the middle of the nineteenth century, small patches of land were irrigated alongside rivers in the western United States of America which expanded throughout the area during the end of the century. This expansion served as a stepping stone for numerous studies on water requirements for agricultural crops, followed by the implementation of multiple mechanisms in order to permit the research of water requirements for various agricultural crops in the United States of America [69].

One of the popular mechanisms that was employed is known as the *Hatch act* [69], which was passed in 1887. This act stated that research may take place at a number of experimental stations located in the Western part of the United States of America, which is now known as New Mexico. The studies that were conducted at these stations frequently involved the assessment of water demand for agricultural crops. Thereafter, the term *duty of water* was extensively used to describe the amount of water used for irrigation purposes during the period 1890 to 1920. During this period, field and plot studies were also implemented in order to determine the relationship between crop yield return, yield losses due to evaporation and the quantity of water used to irrigate crops.

According to Jensen [69], the most widely recognised investigation during the period 1890 to 1920 was the study done by Briggs and Shantz [11, 12] on plant transpiration. The aim of the study was to determine the relative water requirements of crops as a result of transpiration, where meteorological factors[1] play an important role in the rate of transpiration. From the study of Briggs and Shantz [11, 12], multiple other investigations followed with respect to the effect of meteorological factors on evaporation and transpiration. In conclusion, the aforementioned short summary of the research done on irrigation and meteorological factors in the United States resulted in multiple studies on crop water requirements, the measuring of water delivery to farms, studies on factors affecting and causing water losses, the development of procedures in order to estimate seasonal consumptive use of water, the relationship between crop yield and water use, and the formulation of the Penman equation to compute the reference evapotranspiration.

As a result of the multiple studies done on crop development and crop irrigation, it is now possible to estimate the effect of water shortages on crop final yield using mathematical yield predicting formulae. These formulas incorporate the notion of evapotranspiration, and is briefly discussed in the following section.

2.2 The notions of evaporation, transpiration and evapotranspiration

The notion of *evapotranspiration* shares fundamental properties with three other notions, namely *precipitation*, *evaporation*, and *transpiration*. Evapotranspiration plays an important role in crop water demands and governs the amount of yield produced by a crop. In this section, the

[1] Meteorological factors refer to climatological and physical parameters, some of which are measured at weather stations. These factors provide energy in the environment which may result in the removal of water vapour from evaporative surfaces as well as vaporisation [2].

aforementioned notions are thoroughly discussed, as well as multiple environmental conditions that contribute towards the rate at which evapotranspiration takes place.

In nature, precipitation is responsible for condensation from atmospheric water vapour that falls to the ground as a result of gravity [3]. From here, water is extracted from the soil by crop roots and enters the crop pores[2] in the foliage via crop stems. The function of precipitation is to supply moisture to the soil as well as crop surfaces. On the other hand, evaporation is known as the process where water is converted to vapour and escapes into the atmosphere. In the case of irrigating crops, evaporation refers to water that is vaporised from the soil and leaves vegetation, and only occurs as a result of energy in the environment [13]. Finally, transpiration is known as the process where water is converted to vapour from the tiny pores in the foliage, transported via the roots, and lost in the atmosphere [13]. This results in more water to be extracted from the soil by the crop as transpiration continuously occurs. The only difference, however, between evaporation and transpiration is the way in which water is removed from vegetative surfaces, since evaporation directly transports the water from the soil while transpiration directly transports water from crop surfaces.

The notion of evapotranspiration is born, and is denoted as the process whereby water is lost from vegetative surfaces as a combination of soil evaporation and plant transpiration [13]. Furthermore, evapotranspiration may only occur if sufficient water is present in the soil, and therefore, precipitation is regarded as part of the evapotranspiration process. The notions of evaporation, transpiration and precipitation are illustrated graphically in Figure 2.1.

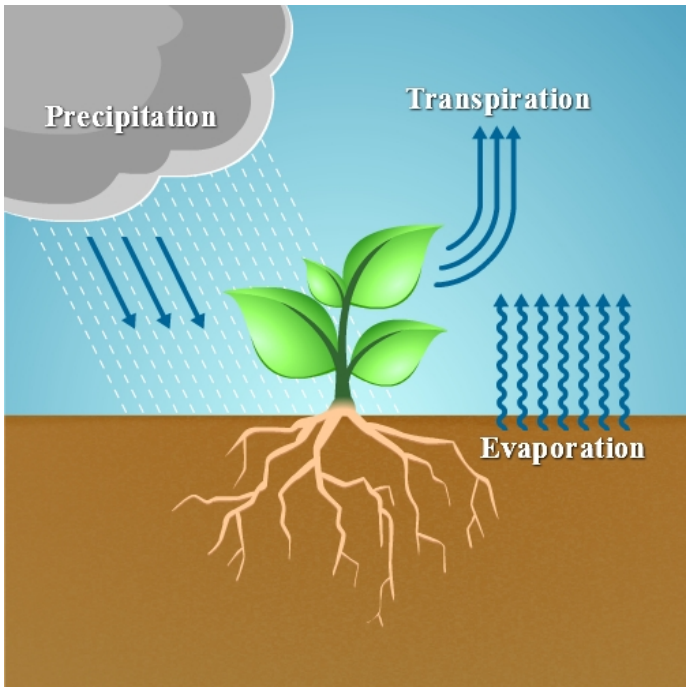


FIGURE 2.1: *The notions of precipitation, evaporation, and transpiration [13].*

The rate at which evapotranspiration occurs is determined by four critical environmental (vegetation) factors [13]. The first factor is soil moisture and is regarded as the most critical factor of the four. Soil moisture simply refers to the water that is held between the soil particles, and as a result, evapotranspiration cannot take place if sufficient water is not present in the soil or crop limbs [42]. The remaining three factors, therefore, depend on the availability of moisture

[2] Also known as the *stomata*, pores are defined as tiny openings in plant leaves that allows for the exchange of gasses. A typical form of gas exchange is the collection of carbon dioxide for photosynthesis [128].

in the soil. The second factor is the crop type which also greatly affects the evapotranspiration rate. One example of this is the fact that grass and other non-native plants requires a considerable amount of water when grown in the desert compared to native crops, which requires less water [13]. This is due to the physical properties that each plant exhibits. Furthermore, the third factor is the respective growth stage that a crop is in. Crops are typically associated with a number of growth stages during a crop's life cycle, and the demand for water during these growth stages varies for different crop types [35]. Higher crop water demands may result in more frequent irrigation, and thus contributes towards higher soil moisture levels and ultimately affects the evapotranspiration rate. The fourth and final factor is weather conditions. As previously mentioned, for evapotranspiration to take place, energy is required in the environment. According to Brown [13], weather conditions are responsible for dictating the amount of energy in the environment, and therefore, plays an important role in the process of evapotranspiration.

Further elaboration on weather conditions resulted in four additional parameters that affects weather conditions. These parameters include solar radiation, humidity, wind speed and temperature. Solar radiation supplies large amounts of energy to the environment in the form of heat, and therefore, is considered to have the biggest impact on the evapotranspiration rate amongst the remaining weather parameters. The wind parameter is considered the second most influential factor, since heat is transferred from adjacent surfaces to vegetation, and water vapour is transferred from moist vegetation to dry atmosphere. Moreover, humidity and temperature work in unity with one another and is responsible for the amount of drying power in the atmosphere, which is quantifiable by means of the meteorological variable called the *vapour pressure deficit*[3] (VPD). Finally, as previously mentioned, the temperature impacts the VPD and also contributes towards higher energy in the environment. Considering that all factors are equal, the evapotranspiration rate will be higher at a warmer vegetation temperature compared to a lower vegetation temperature since less energy is required in order to evaporate water from warming vegetation surfaces [13].

During the process of evapotranspiration, soil evaporation and crop transpiration occurs simultaneously. Considering that when atmospheric conditions and irrigation practices are kept constant, evapotranspiration rates may still vary as the crop's physical characteristics change throughout it's growing period. Apart from water availability in the topsoil, the fraction of radiation that reaches the soil surface determines the evaporation. When a crop is small, the predominant factor responsible for transferring moisture from vegetation is evaporation. When a crop is well developed, transpiration becomes the predominant factor since expanded crop canopies result in solar radiation to reach crop foliage rather than the soil covered in shade [2]. In Figure 2.2, the changing fractions of evaporation and transpiration occurring simultaneously throughout the growing period of a crop are illustrated. Furthermore, the *leaf area index* (LAI) denotes the total amount of foliage area covering the ground surface (in $\frac{m^2}{m^2}$). As seen from Figure 2.2, 100% of the evapotranspiration taking place during sowing is due to evaporation, whereas at full crop cover (*i.e.* highest LAI), transpiration accounts for 80% of the total evapotranspiration taking place.

The evapotranspiration may be measured using specific physical appliances such as, for example, an evapotranspiration monitoring system in combination with corresponding methods, or may be computed using a number of mathematical methods. The latter method include an energy balance method, microclimatological methods, and a soil water balance. These type of methods are often expensive, demanding in terms of the accuracy of measurements, and requires well-trained research personnel in order to fully exploit these methods. On the other hand, the

[3]The VPD estimates the difference between moist vegetation and drier atmosphere above the vegetation, also known as the concentration of water vapour [13].

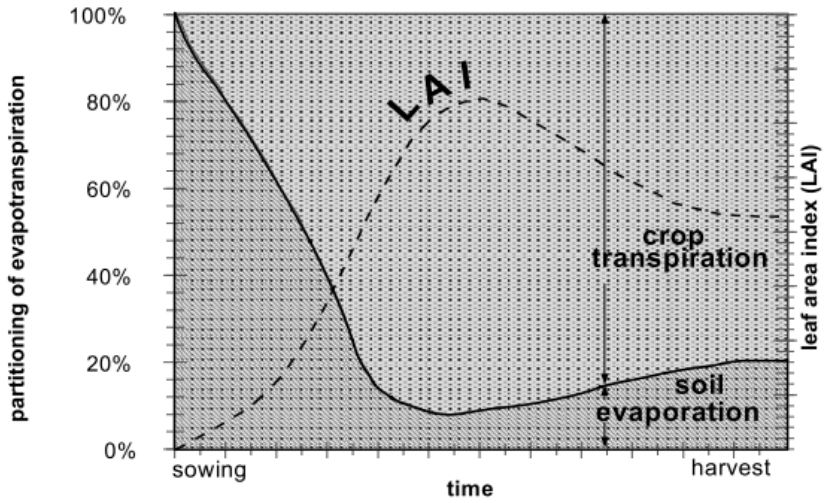


FIGURE 2.2: *The partitioning of evapotranspiration from the time of sowing till the time of harvest, while the LAI changes as the crop progresses through it's life cycle [2].*

evaporation/evapotranspiration may be computed using meteorological data with corresponding equations. The Penman-Monteith equation is regarded as the standard method for computing the reference evapotranspiration when using meteorological data [2].

In conclusion, evapotranspiration is responsible for reducing soil and crop moisture, and is therefore an important aspect to consider when quantifying crop water demands. Moreover, the effect of evaporation on reservoir water surfaces is also noteworthy. More frequently, reservoirs are used for irrigation purposes, and is exposed to significant amounts of radiation during summer seasons. In the following section the estimation of evaporation using reservoir characteristics and historic pan evaporative data are briefly described.

2.3 Estimating evaporation from water surface areas

As previously defined, evaporation is responsible for transferring water into the atmosphere, and may also include transferring water from reservoir water surfaces. There are a number of methods that vary in accuracy for estimating evaporation losses from open-air reservoirs. In this section, a detailed review is given as an overview on some of these existing evaporation loss methods, followed by a discussion on a model approach to predict future evaporation losses.

There are three categories in which evaporation estimation methods may be divided into, namely comparative methods, budget methods and aerodynamic methods [145]. In a comparative method, the actual evaporation is physically measured from evaporation pans in a close proximity to the reservoir, which is then used to estimate the corresponding expected actual evaporation from the reservoir. The accuracy of this method, however, is dependent on the ability of the instrument to mirror the effects of wind and heat absorption characteristics that the reservoir may experience daily [145]. In a budget method approach, a balance equation is employed in order to estimate the past evaporation losses by means of equating reservoir outflows to inflows, and adding the change in storage level as well as the evaporation loss. From here, the inflows, outflows and changing water level can be physically measured, which enables the reader to estimate the remaining evaporation loss for a given time period. All reservoir inflows and outflows, as well as the seepage rate through the reservoir wall, should be measured as accurately as possible, since the model accuracy is highly dependent on these measures. An example of this

class is the energy balance method and the water budget method [145]. Finally, an aerodynamic method includes a subset of three other methods, namely mass transfer, eddy correlation, and gradient methods. These methods are used to estimate the evaporation loss on the water surface area by means of the heat and humidity distributions, and the air velocity [145].

For the comparative approach, Sun *et al.* [124] stated that there exists a proportional relationship between the evaporation loss and the exposed water surface area. Consequently, Van Vuuren and Gruñdlingh [137] modelled the evaporation losses as a function of the exposed surface area, and is expressed as

$$Ev_i = \upsilon_i \times \frac{(A_i + A_{i+1})}{2}, \tag{2.1}$$

where Ev_i denotes the volume of evaporation taking place during calculation period i, υ_i denotes the rate of evaporation related to the calculation period i, and A_i and A_{i+1} denote the exposed water surface area for the calculation period i and $(i-1)$, respectively. Van Vuuren and Gruñdlingh [137] assumed a linear relationship between the reservoir volume and exposed water surface area as a result of a linear programming model that was used. Considering the topography that surrounds multiple reservoirs, the reservoir volume and exposed water surface area relationship is rather considered non-linear more frequently. Sun *et al.* [124] concluded that a piecewise linear approximation may be fitted to a non-linear relationship in the case where the latter is true. This is done in order to obtain a mathematical function which represents a reservoir shape characteristic such that the exposed reservoir surface area is available at any reservoir volume.

2.4 Estimating crop final yield

There are a number of mathematical functions which relate crop production with water use, and these functions enables scientists to estimate crop water requirements and the effect that water deficits have on crop final yield. These mathematical functions requires are a number of dependent and independent variables as well as a number of parameters. In this section, these parameters and variables are thoroughly described, followed by a step-by-step procedure to evaluate the accuracy of these parameters as well as methods to obtain these parameters. Moreover, the mathematical functions to predict crop final yield are discussed, followed by a brief overview of multiple DSSs that incorporate such functions for similar purposes. This section starts with a brief description on crop growth stages.

2.4.1 Crop growth stages

From the initial planting until the harvesting stage, a crop exhibits a number of physical changes which relates to crop height, ground cover and leaf area. Each distinctive physical attribute may be grouped into stages known as crop growth stages. There are four distinct growth stages associated with the growing of a crop, and include (1) the initial stage, (2) the crop development stage, (3) the mid-season stage and (4) the late season stage [2, 35]. Each respective growth stage contributes to various rates of evapotranspiration as a result of distinct physical attributes, as discussed in §2.2.

The first growth stage starts when the crop is planted and lasts until approximately 10% of the ground is covered by green vegetation (*i.e.* when LAI = 10%). The time window of this stage is dependent on the respective crop and it's variety, the date of planting, and climatic conditions. For perennial[4] crops in the following year, the initiation of new foliage represents the starting date. The crop development stage, on the other hand, starts when LAI = 10% ground cover and lasts until effective full ground cover (*i.e.* when LAI = 100%) is achieved. For many crops, the effective full cover occurs at the time of flowering. For crops that are grown in rows, effective full cover starts when leaves are interlocking and the soil between these crops are fully shaded, or when the crop reaches full size if no interlocking takes place. Examples of such crops include potatoes, sugar beets, corn, beans, to name a few. Furthermore, for some crops that grow taller than 0.5 m, the LAI may be used to determine the effective full cover and is usually approximately $70\% - 80\%$ [2]. It is, however, difficult to visually determine effective full cover for crops that are densely sown, and therefore, the start of flowering is used as the effective full cover for such crops.

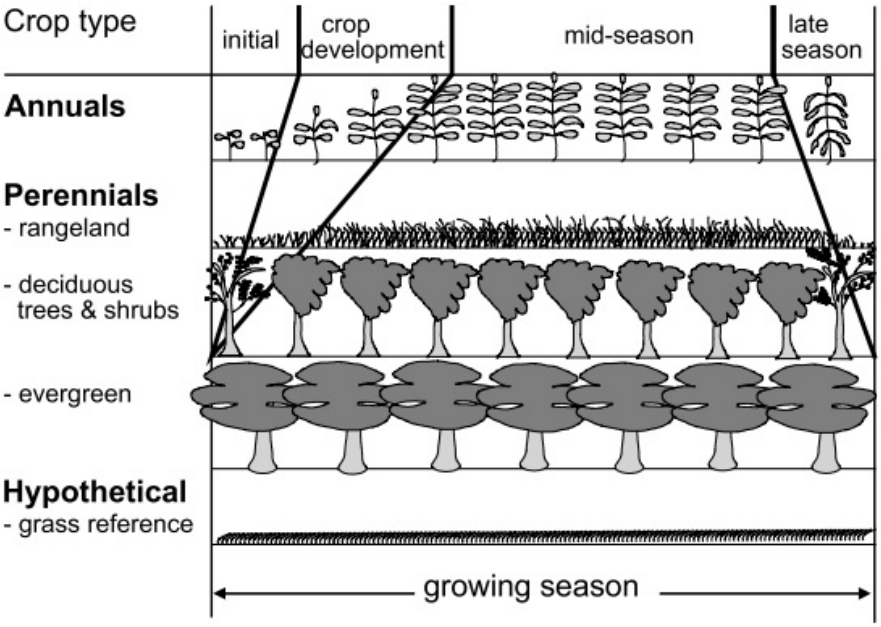


FIGURE 2.3: *The four distinct growth stages in the life cycle of annual crops, perennial crops and a hypothetical grass reference surface. The growing season marks the life cycle of each respective crop type [2].*

Moreover, the mid-season stage starts from effective full cover and stretches until the crop reaches maturity. The following physical attributes normally indicate when a crop has reached maturity: the senescence or yellowing of foliage, the dropping of foliage, or the browning of fruit. For perennial crops, as well as a number of annual crops, the mid-season stage are regarded as the longest time period. Finally, the late season stage starts from the crop maturity and ends when the crop is harvested or dried out. For perennials, it can be assumed that the end date is equal to the date of planting. The physical crop development of annual crops, perennial crops, and the hypothetical grass reference surface, for the distinct growth stages (1)–(4), are illustrated in Figure 2.3. The hypothetical grass reference surface is used to determine the reference evapotranspiration in order to study the evaporative demand of atmospheric conditions independently of crop type, crop development and management practices. From Figure 2.3, it is

[4] A term used to differentiate between the lifetime of crops, and denotes, in this case, crops that grow for more than two years [127].

clear that perennial crops do not exhibit aggressive physical changes compared to annual crops, whereas the hypothetical grass reference surface exhibits no change in physical attributes. In order to determine the total evapotranspiration, a crop coefficient may be jointly used with crop growth stages to determine the evapotranspiration for each growth stage and then, in an additive manner, estimate the total evapotranspiration. The notion of crop coefficients and yield response factors are briefly described in the following section.

2.4.2 The crop coefficient and yield response factor

In 1979, Doorenbos and Kassam [35] proposed a relationship that relates the reduction in actual yield with the reduction in the relative evapotranspiration by taking into account a crop coefficient and yield response factor. These two notions were the solution to complexed relationships encountered between crop, soil, water and climatic conditions, which were reduced to manageable components that allowed for the analysis of crop yield response towards water deficits. There is a noticeable difference between crop coefficients and yield response factors, which is subsequently described in the remainder of this section.

In order to describe the role that crop coefficients and yield response factors play in predicting the final crop yield, it is important to understand the number of aspects that affects crop evapotranspiration. As mentioned in §2.2, climatic conditions and crop characteristics play an important role in the total evapotranspiration taking place. According to Allen [2], the computation of total evapotranspiration using meteorological data is estimated in two stages. The first stage entails the implementation of a grass reference surface, defined in §2.4.1, under optimal environmental conditions in order to determine a so-called *reference crop evapotranspiration*. The aim of this method is to study the evaporative demand from a reference surface independently of crop characteristics such as crop type, crop development, and management practices, and is only affected by climatic conditions, namely radiation, wind speed, temperature and humidity. In an optimal environment, water is abundant which ensures that the evapotranspiration is not affected by soil factors. Furthermore, by relating the evapotranspiration to a specific surface, a reference is established which relates evapotranspiration to other surfaces as well [2].

The reference crop evapotranspiration is denoted as ET_o, and may be calculated by means of the so called Penman-Monteith equation. This equation requires a number of climatic parameters that are measured by weather stations world wide. Moreover, this equation as well as the Penman equation are also the most widely used methods today when it comes to estimating the ET_o [13]. According to the Irrigation and Drainage paper No. 56 "Crop evapotranspiration" [2], the ET_o is calculated as follows

$$ET_o = \frac{0.408\alpha\,(R_n - P) + \rho\left(\frac{900}{T+273}\right)u_2(e_s - e_a)}{\alpha + \rho\,(1 + 0.34u_2)}, \tag{2.2}$$

where α denotes the slope of the vapour pressure curve (kPa°C^{-1}), R_n denotes the net radiation at the surface of the crop (MJ m^{-2}day^{-1}), and P denotes the density of the soil heat flux (MJ m^{-2}day^{-1}). Furthermore, ρ denotes the psychometric constant (kPa°C^{-1}), T denotes the average daily temperature at a height of 2m (°C), u_2 denotes the wind speed at a heigh of 2m (ms^{-1}), and finally, $(e_s - e_a)$ denotes the saturation vapour pressure deficit (kPa). These parameters, however, will not be thoroughly discussed since estimating the ET_o is now possible using a number of DSSs available online, rather than estimating the ET_o by hand. The only requirement is to select a weather station where the meteorological data will then be automatically gathered and inserted into equation (2.2). Moreover, a small variation in the estimated

ET_o values across large areas are considered highly dubious, and therefore useless. According to Brown [13], however, the ET_o values are surprisingly stable over large areas, since solar radiation has the biggest impact on the ET_o, as mentioned in §2.2, and is considered stable over large areas. Furthermore, the temperature, humidity and wind speed have a rather small impact on ET_o values, thus suggesting that small variations in parameter values may be disregarded in this case [13].

The second stage involves the estimation of crop evapotranspiration and is denoted as ET_c. This estimation incorporates the ET_o in conjunction with a crop coefficient, which is denoted as K_c. The crop coefficient integrates the effects of characteristics between field crops and hypothetical grass reference surface into a single coefficient. This approach is referred to as the crop coefficient approach for estimating crop evapotranspiration, and simply entails multiplying the K_c coefficient with the estimated ET_o to compute the ET_c. Furthermore, K_c varies predominately with specific crop characteristics, but also with a limited extent of climatic conditions. It is, therefore, possible to transfer standard values of K_c between climates and locations which leads to the computation of crop evapotranspiration at various locations [2]. The procedure for estimating the ET_c using the crop coefficient approach is illustrated in Figure 2.4.

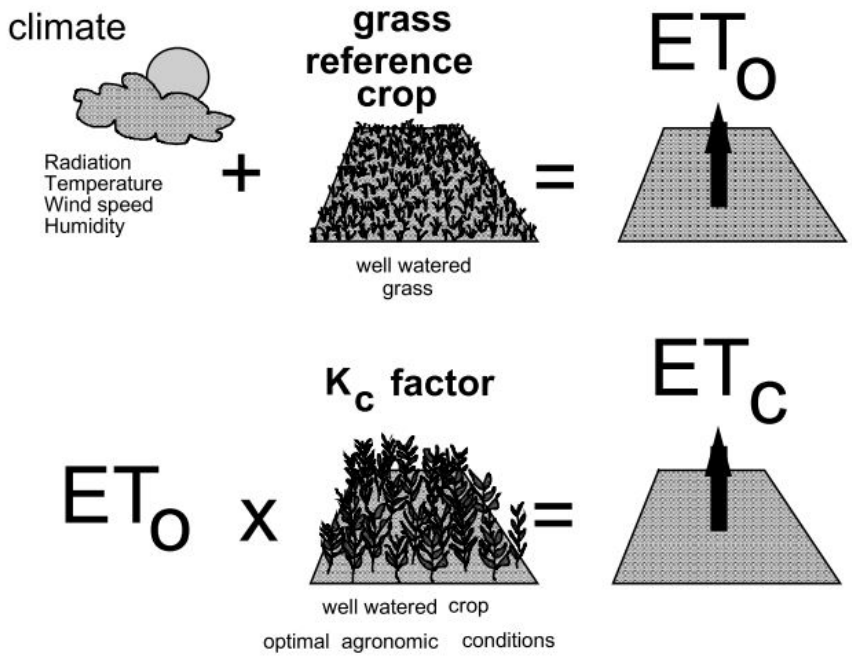


FIGURE 2.4: *Estimating the ET_c using the crop coefficient approach. This entails incorporating climatic conditions and a crop coefficiënt K_c [118].*

Integrated in the crop coefficient are four characteristics that distinguish a crop from the grass reference surface, and include the crop height, the reflectance (albedo) of crop-soil surface, the evaporation taking place from the soil, and the canopy resistance [2]. The crop height impacts the turbulent vapour transfer rate from the crop into the atmosphere, and also the aerodynamic resistance. The crop-soil surface reflection, on the other hand, impacts the net radiation (R_n), as denoted in equation (2.2), as a result of vegetation covering the ground and wet soil surfaces. Finally, the canopy resistance impacts the surface resistance of a crop towards vapour transfer and is affected by the age, area and condition of the foliage.

The changing physical characteristics of a crop, as well as the environmental characteristics, also have an impact on the K_c coefficient. These characteristics include the crop type, the climatic conditions, the evaporation from the soil, and the crop growth stages as explained in §2.4.1. As previously mentioned, the K_c coefficient corresponds with each growth stage in a crop life cycle, which leads to the estimation of crop evapotranspiration during each respective growth stage. Referring to the growth stages in §2.4.1, the K_c coincide with soil moisture content during the initial stage since crop vegetation is almost non-existing. This concludes that the K_c is high when soil is wet and low when soil is dry, and ultimately drawing inference that the K_c is a function of the frequency of watering (*i.e.* a more frequently wet soil moisture may result in a high K_c value). For the development stage, the K_c corresponds with the amount of ground covered by vegetation, and concludes to a relatively dry soil moisture content, where $K_c = 0.5$ for $25 - 40\%$ ground cover and $K_c = 0.7$ for $40 - 60\%$ ground cover. These values, however, will vary depending on the crop type, frequency of watering, and the crop water requirements when compared to the reference crop at full ground cover [2]. For the mid-season stage, the K_c is considered relatively constant for most growing conditions, and also reaches it's maximum value. For the late season stage, the K_c value is dependent on the frequency of watering demanded by the respective crop until harvested [2]. In Figure 2.5, the band width for the K_c coefficient across four growth stages for sugar cane, cotton, maize, cabbage, onions and apples are illustrated.

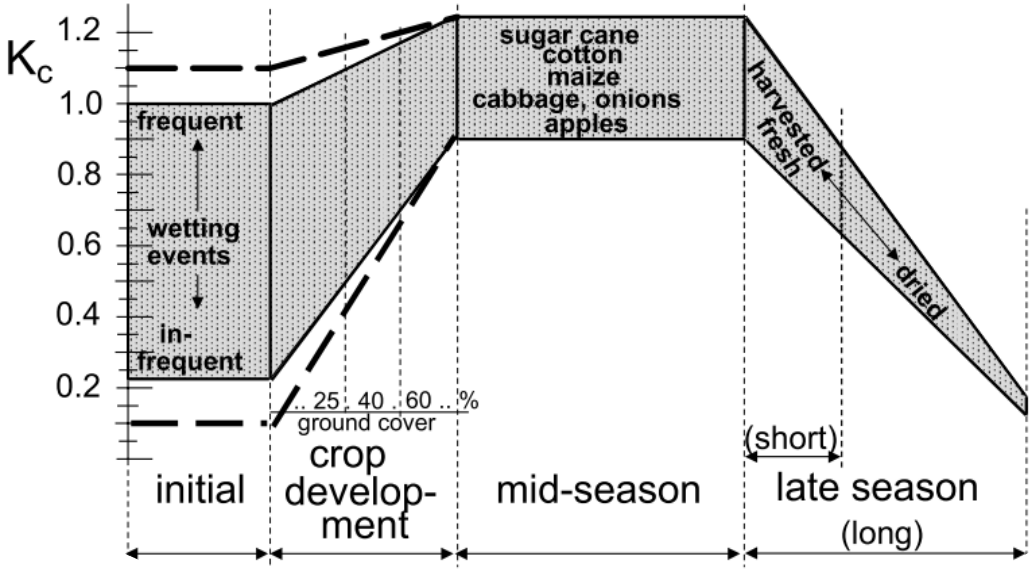


FIGURE 2.5: *The typical ranges for K_c values across four growth stages for sugar cane, cotton, maize, cabbage, onions and apples [2].*

When considering the yield response factor, denoted as K_y, a complex linkage between final yield and water irrigation to the crop is captured into a single factor. The aim of this factor is to describe the sensitivity of a crop towards a water deficit, and is used in mathematical crop water production functions in order to asses the yield reduction as a result of inadequate irrigation. According to Doorenbos and Kassam [35], a water deficit is presented as the ratio of actual evapotranspiration to the maximum evapotranspiration, where the former is denoted as ET_a and the latter is denoted as ET_m. This ratio may also be referred to as the ratio of $\frac{\text{water supply}}{\text{water demand}}$ as assumed by Wardlaw and Barnes [140], and is referred to as the *evapotranspiration* ratio in the remainder of this thesis. From the ratio of Wardlaw and Barnes [140], the crop water demand corresponds with the total crop evapotranspiration ET_c. Moreover, the magnitude

of a water deficit may be assessed with respect to the crop water requirements over the total growing season, or in with respect to crop water requirements of individual growth periods. These individual growth periods, on multiple occasions, do not correspond with the growth stages discussed in §2.4.1 in terms of the duration and, are therefore, labelled according to their respective physical appearances. These growth periods include (0) the establishment period, (1) the vegetative period, (2) the flowering period, (3) the yield formation period, and (4) the ripening period. Finally, a K_y factor is assigned to each growth period in order to asses the total yield reduction for the total growing season [35].

The derived K_y values are based on the assumption that the relationship between the relative yield and the relative evapotranspiration is linear, and is only valid if this ratio is smaller or equal to 50%. A general relationship between the relative yield and the relative evapotranspiration was proposed by Doorenbos and Kassam [35] in 1979, and is expressed as

$$\left(1 - \frac{Y_a}{Y_m}\right) = K_y \left(1 - \frac{ET_a}{ET_m}\right), \tag{2.3}$$

where Y_a denotes the actual yield, Y_m denotes the maximum yield, and the remaining parameters are as previously defined. This relationship describes the reduction in yield as a result of water deficit by means of a yield response factor. According to Doorenbos and Kassam [35], the ratio may not exceed the following condition, as stated previously, and is expressed as

$$0.0 \leq \left(1 - \frac{ET_a}{ET_m}\right) \leq 0.5. \tag{2.4}$$

The condition in (2.4) may be adjusted to obtain a range of evapotranspiration ratios which indicates to what extent this ratio may be varied in order for the assumption to hold. The lower and upper bounds are estimated from equation (2.4) to obtain

$$0.5 \leq \frac{ET_a}{ET_m} \leq 1.0. \tag{2.5}$$

The condition in (2.5) simply explain that the water supplied to the crop may not exceed the crop water demand (100%), and may not be lower than half of the crop water demand (50%). Further elaboration on the relationship in equation (2.3) will be discussed later on in this chapter, and is only mentioned in this section for explanatory purposes.

For each K_y factor, an associated value is assigned which indicates the sensitivity of a respective growth period of a crop towards a water deficit. A value of $K_y > 1$ indicates a highly sensitive response to water deficit and may result in gradual reductions in crop yields within a constrained water environment. A value of $K_y < 1$ indicate that crops are less sensitive to water deficit and is expected to recover easily from partial water stress. A value of $K_y = 1$ indicates that the ratio of yield reduction is directly proportional to the evapotranspiration ratio [2, 35, 118]. Furthermore, Steduto *et al.* [118] illustrated the linear relationship between the relative yield reduction and relative evapotranspiration reduction for maize for a number of growth periods. The y-axis denotes the relative yield reduction and the x-axis denotes the relative evapotranspiration reduction, and is illustrated in Figure 2.6. In this Figure, the yield response factors are shown for all the growth periods, and additionally, for growth periods (1)–(2) and (2)–(3). If the gradient (shown as the blue line in Figure 2.6) tends more towards the y-axis, the growth period is more sensitive to inadequate water supply and takes on a high K_y value. If, however, the gradient tends more towards the x-axis, the growth period is less sensitive to inadequate water supply and takes on a lower K_y value, as illustrated in Figure 2.6.

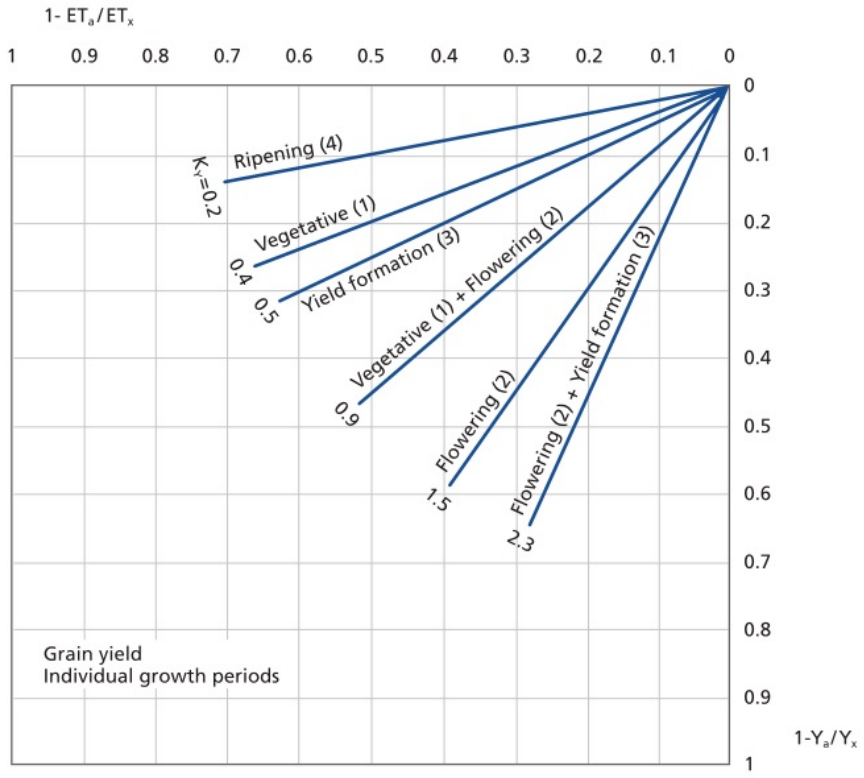


FIGURE 2.6: *The linear relationship between the relative yield reduction and relative evapotranspiration reduction for a number of growth periods, as proposed by Steduto et al. [118]. Additional K_y values were also derived for the combination of growth periods (1)–(2) and (2)–(3).*

2.4.3 The maximum yield of crops

The maximum yield parameter plays an important role in estimating the final crop yield. Doorenbos and Kassam [35] proposed a different approach to calculate the crop coefficient which is briefly described in this section.

According to Doorenbos and Kassam [35], the maximum yield is defined as the yield harvested from a high-producing crop variety that is grown in an optimal environment (*i.e.* water, nutrients, pests and diseases are not a limiting factor in the yield production). Considering climatic conditions, on the other hand, there are a number of factors that will affect the maximum yield, which includes the temperature, the exposure to radiation, the total length of the growing season regulated by the daily temperature, and the required day-length for crop development [35]. The temperature determines the crop development rate and thus affects the crop's ability to form yield. It also affects the quality of the yield produced by the crop. Moreover, radiation is responsible for affecting the crop growth and the crop yield since radiation and temperature coincide with one another. Furthermore, multiple crops have varying growth periods, and therefore require different climatic conditions in order to reach their full potential, respectively. Crops, however, have been able to adapt to a wide range of climatic conditions as well as growing crops outside its original growing period [35]. For this reason, the maximum yield is considered a function of climatic conditions.

Doorenbos and Kassam [35] proposed two computational techniques that may be used to estimate the maximum yield for different climatic conditions, namely the *Wageningen method* and the *Agro-ecological zone method*. These methods, however, are considered to be too com-

plex since the interrelationships between a number of parameters makes the derivation of these methods too complicated [35]. On the other hand, an optimal environment is seldom achieved since drought conditions, pests and a lack of ground nutrients are evident in today's farming environments. It is, therefore, suggested that the maximum yield parameter be replaced with an average estimation of the yield gained over a number of years preceding — there are two clear advantages of taking this approach.

The first advantage includes the underestimation of the actual yield, which may ultimately benefit the farmer since a lower estimation of actual yield caries less risk than a higher estimation. The second advantage includes a more accurate estimation of the actual yield since a population of data is used to determine the average yield. Therefore, the actual yield is not reliant on a single value obtained for a number of years of farming. In a practical sense, a reduction from the average yield obtained from the equilibrium state farming practices is considered more practical compared to a reduction from the maximum yield that was obtained only once in the history of the farm.

To estimate the average yield for crops, two additional methods may be considered to do so. The first method entails estimating the average yield of crops using historical data obtained for a specific crop for a specific location. The FAOSTAT website [51] may be used in this case to obtain historical data for a number of crops for a number of locations. The aim of this approach is to capture an average estimation which incorporates location specific data over a number of years, and may include various climatic conditions, different soil types and different growing conditions. The second method refers to computing the average yield from data recorder by the farmer based on the historical yields that was obtained by the crops grown on his/her farm.

When visiting the FAOSTAT website, the expected webpage to be encountered is illustrated in Figure 2.7. Next to the 🏠 tab in the left corner, the "Data" tab must be selected, and under the "Production" title, "Crops" must be selected, as illustrated in Figure 2.7. After selections have been made, the following interface in Figure 2.8 can be expected. From here, the specific country may be selected from the left, various crops from the bottom left, the required elements to be displayed for each crop from the top right, and the required number of years from the bottom right. After the required elements have been chosen, the "Download Data" tab may be selected at the bottom of the page in order to download the data in an Excel spreadsheet format.

After the data have been downloaded, the average ton per hectare for maize for each year may be calculated by dividing the production quantity by the number of area harvested for each year, which is indicated by two of the three elements selected in Figure 2.8. Furthermore, the estimated average ton per hectares captured in the downloaded *Excel* spreadsheet for the selected years 1999–2016 for maize is illustrated in Table 2.1. It can be seen that from years 2008–2016, the average ton per hectare increased in comparison to the years 1999–2007, resulting in an average of 4.481 ton per hectare for years 2008–2016 and an average of 2.877 ton per hectare for years 1999–2007. This occurrence may be due to more efficient practices that were implemented at the start of the year 2008.

The question remains, however, how many years of data should be used from the FAOSTAT database to ensure that the calculated average is regarded as a good average estimation. This yields that if the standard deviation is one order smaller than the average, it may be regarded as a good average [136]. For example, the average ton per hectare for years 1999–2007 yielded 2.877, of which the standard deviation yielded 0.287 ton per hectare. Since the current farming practices yield higher maize production, historic data dating back to 2008 may used to estimate the average ton per hectare for relativity purposes. Moreover, the average ton per hectare for years 2008–2016 was estimated as 4.481, and the standard deviation was found to be 0.472

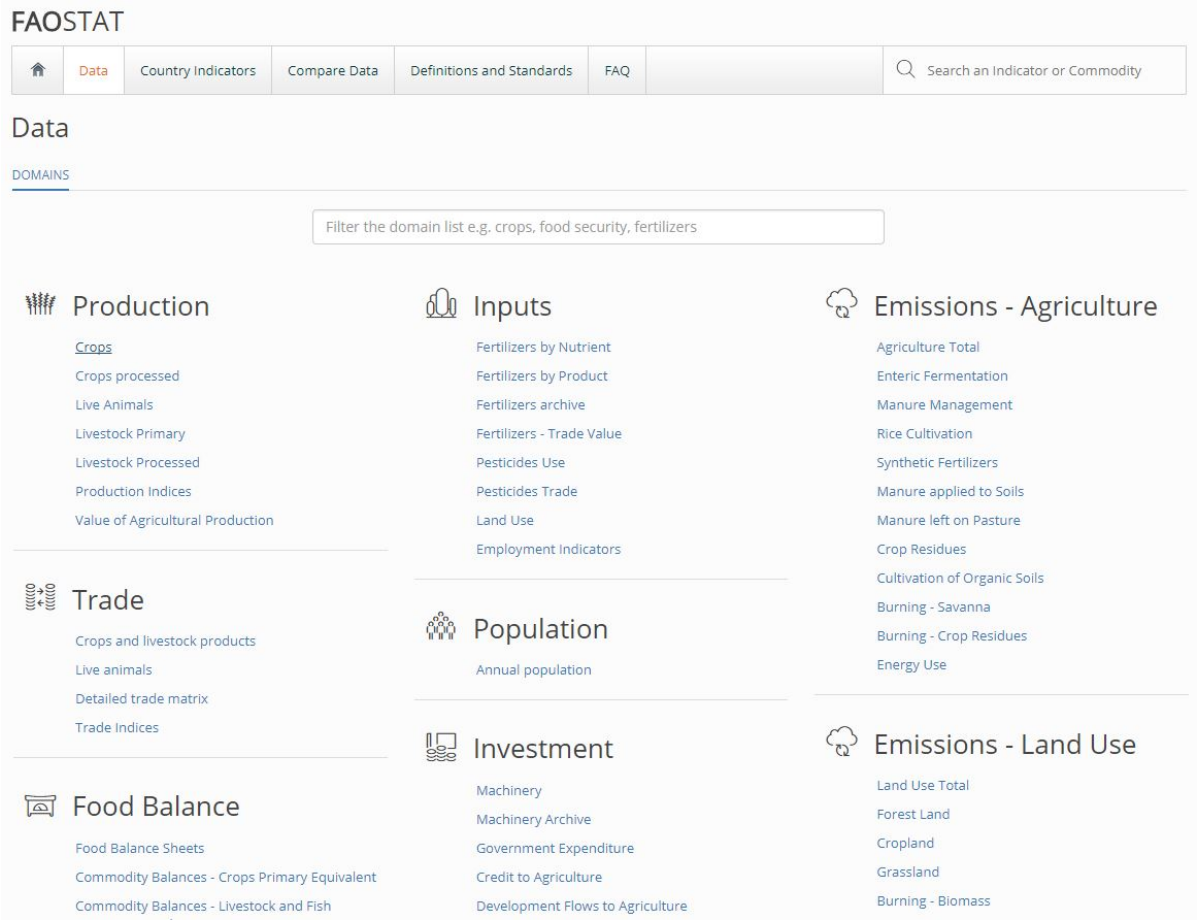


FIGURE 2.7: *The home page of the FAOSTAT website [51]. From this website, information on multiple aspects in the farming environment may be acquired. For this specific purpose, the historical data of crop production were gathered.*

ton per hectare. This concludes that when implementing this approach, the data from 2008 onwards must be used for maize to ensure an accurate estimation of the average and, ultimately, an accurate estimation of final yield production. This approach may also be applied to other crops selectable from FAOSTAT in order to determine a good average. The data acquired from the FAOSTAT website, however, is not location specific and rather indicates the obtained average yield per hectare across a country. For this reason, the method is considered as an alternative way of calculating the average yield production. This method may also be used to validate historical yield records obtained by farmers.

2.4.4 Crop water production functions

The relationship between crop production and water has long been a subject of interest among irrigation, crop and plant enthusiasts. A number of methods have been developed which relates the crop production with water use and facilitates multiple discussions on the aforementioned relationship to a great extend. The methods that were developed may be divided into two groups namely *water production functions* (WPF) and *crop water production functions* (CWPF) [66]. WPF, as named by Kipkorir *et al.* [72], relates crop yield with water application, where a number of processes related to water applications include irrigating crops, precipitation, pre-planting

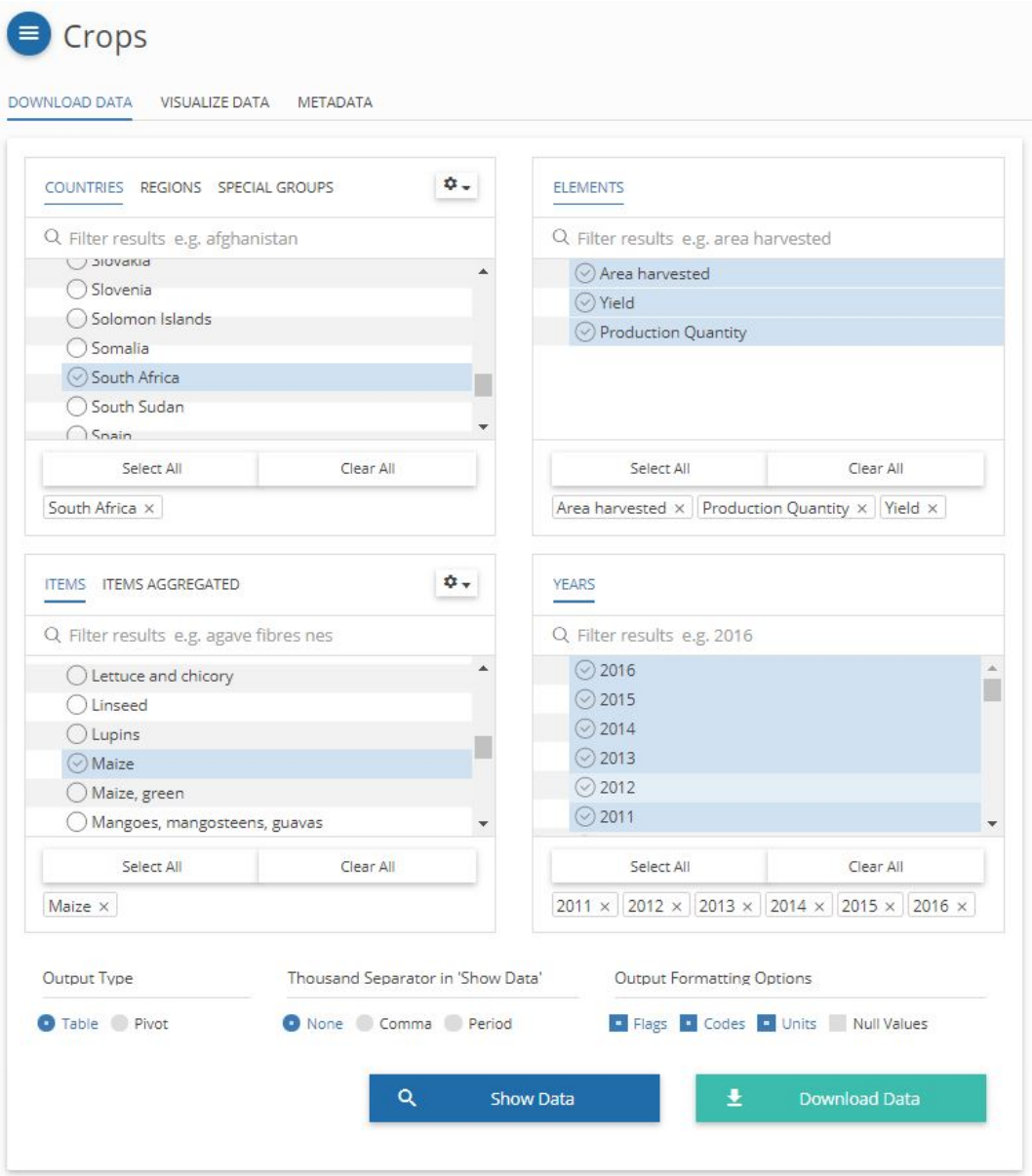


FIGURE 2.8: *Four selection criteria available to the user for selection. This entails selecting country, region or special groups, specific items which are longed for, different elements as well as specific years of the recorder data [51].*

irrigation or leaching requirements[5]. CWPF, on the other hand, relates the crop yield with seasonal evapotranspiration, and is very useful in the field of irrigation water management applications. These applications may be used to evaluate economic implications with respect to water irrigated to crops, and also to determine irrigation strategies when water supply is limited [66]. In the remainder of this section, multiple CWPFs are briefly discussed and critically analysed.

It is important to note that CWPF varies among crop varieties and climatic zones since evapotranspiration rates are affected by the physical characteristics of each respective crop as well as climatic conditions. According to Rhenals and Bras [104], there is no universal CWPF that

[5]The process where the accumulation of excessive salt build up is prevented by applying excess of water to crops. The excess water then passes through the root zone and leaches the excessive soluble salts [20].

Year	Average ton per hectare
1999	2.227
2000	2.849
2001	2.437
2002	2.852
2003	2.658
2004	3.031
2005	3.635
2006	3.412
2007	2.792
2008	4.537
2009	4.964
2010	4.674
2011	4.367
2012	4.490
2013	4.247
2014	5.301
2015	3.753
2016	3.996

TABLE 2.1: *The estimated average ton per hectare for maize for years 1999–2016 downloaded from the FAOSTAT website [51].*

may be applied to all crops, growing seasons and climates since both dependent and independent variables that are required in CWPF vary with respect to crop characteristics and environmental factors. Therefore, the performance of CWPF for various crops and locations are required in order to make accurate estimations of the actual yield. If it is possible to quantify these variables and parameters for specific locations, it may be plausible to conduct the effect on the final yield for multiple crops using a single CWPF. Two types of forms are also incorporated in CWPF. The first include forms that relate crop yield with total seasonal evapotranspiration, and the second include forms that relate crop reduction to a water deficit during a specific period in crop growth. The latter usually coincides with crop growth periods, and is also known as *phenology stages* [66]. Furthermore, these forms also suggest that a water deficit in each growth period has a unique effect on final crop yield, and that the effect of a water deficit at one growth period is also dependent on the other periods [66].

A general equation that relates the relative yield reduction to the relative reduction in evapotranspiration may be derived from the work of Stewart and Hagan [120]. The general form of the equation was first introduced in 1973, and is expressed as

$$\frac{Y_x}{Y_m} = 1 - \beta\,(1 - ET)\,, \tag{2.6}$$

where $\frac{Y_x}{Y_m}$ denotes the relative grain yield, ET denotes the vegetation evaporation ratio, and β denotes the yield-water stress response parameter, which is also known as the yield response factor that was described in §2.4.2. The equation in (2.6) enables the estimation of yield reduction in an individual growth period, and in order to apply this equation in such a manner that the total yield reduction may be estimated for all the growth periods simultaneously, some method of combining the yield reductions is required. Three different types of methods within CWPF have been proposed in the literature, and include *exponential-type* of methods, *additive-type* of methods and *multiplicative-type* of methods. The additive-type methods only became signifi-

cant recently [21]. De Jager [21] was one of the first to evaluate five different ways of combining yield reductions in each growth period. These five methods include three additive-type models that were proposed by De Jager [24], Stewart *et al.* [121] and Stewart and Hagan [120], and a multiplicative-type model proposed by Stewart *et al.* [121] as well as an exponential-type model proposed by Jensen [69].

A field experiment was conducted at Roodeplaat in South Africa in order to measure the accuracy of each CWPF. The final yield was estimated under a number of different irrigation strategies and compared with the measured field data for corresponding irrigation strategies. The field experiment adopted a split-plot design[6] which enabled the implementation of multiple water irrigation strategies for spring wheat. Three of the five models were calibrated using data from the field experiment as well as a trial-and-error method. The β-parameters from Doorenbos and Kassam [35] were initially used and adjusted to produce new β-parameters for the additive-type models of De Jager [24] and Stewart *et al.* [121], and new β-parameters for the exponential-type model of Jensen [69]. For the remaining models, the β-parameters were taken from Doorenbos and Kassam [35].

From the paper of De Jager [21], the first model is an additive one and was proposed by De Jager [24] in 1987. This model is expressed as

$$\frac{Y_x}{Y_m} = 1 - \sum_{g=1}^{G} \beta_g \left(1 - \left(\frac{ET_a}{ET_m}\right)_g\right), \tag{2.7}$$

where Y_x denotes the actual yield, Y_m denotes the maximum yield, and β is as previously defined. Moreover, g denotes a specific crop growth period, implying that a yield response factor is assigned to each growth period through β_g, and $\frac{ET_a}{ET_m}$ denotes the evapotranspiration ratio as defined in §2.4.2 and corresponds with the vegetation evaporation ratio ET in equation (2.6). The second model is the model proposed by Stewart *et al.* [121], and makes use of a multiplicative-type of method to combine the yield reductions over all the growth periods. This model is expressed as

$$\frac{Y_x}{Y_m} = \prod_{g=1}^{G} \left(1 - \beta_g \left(1 - \left(\frac{ET_a}{ET_m}\right)_g\right)\right), \tag{2.8}$$

where all the parameters are as previously defined. The third model is known as a *full season* (FS) model and is proposed by Stewart and Hagan [120]. The response factor β is not confined to a specific growth period, but rather takes on a single value for the entire growth season. This model is expressed as

$$\frac{Y_x}{Y_m} = 1 - \beta \left(1 - \sum_{g=1}^{G} \left(\frac{ET_a}{ET_m}\right)_g\right), \tag{2.9}$$

where all the parameters are as previously defined in equation (2.7). The model in equation (2.9) is merely a mirror of the relationship proposed by Stewart and Hagan [120] in equation (2.6). However, the evapotranspiration ratios varies during each growth period while the yield response factor remains constant throughout the growing period. The fourth model is proposed by Stewart

[6] A type of field design where replications of treatments are assigned randomly to various experimental subjects in order to study the various outcomes on the subjects [125].

et al. [121] and makes use of an additive-type of method to combine the yield reductions. This model is expressed as

$$\frac{Y_x}{Y_m} = 1 - \left(\frac{\sum_{g=1}^{G} \beta_g \left(ET_{m_g} - ET_{a_g} \right)}{\sum_{g=1}^{G} ET_{m_g}} \right), \tag{2.10}$$

where ET_{a_g} denotes the actual evapotranspiration for growth period g and ET_{m_g} denotes the maximum evapotranspiration for growth period g. The remaining parameters are as previously defined in equation (2.7). The last model is proposed by Jensen [69] and is considered as the oldest CWPF. This model was proposed in 1968 and makes use of a multiplicative-type of method to combine the yield reductions. This model is expressed as

$$\frac{Y_x}{Y_m} = \prod_{g=1}^{G} \left(\frac{ET_a}{ET_m} \right)^{\beta_g}, \tag{2.11}$$

where all the parameters are as previously defined in equation (2.7).

De Jager [21] used six core validation characteristics in order to model the accuracy of each model. The estimated actual yield gained from different water irrigation strategies was compared to the measured yield harvested from the field experiments for each model, and followed by a statistical analysis on the results obtained. The core validation characteristic includes a slope through the origin fitted on the estimated data points, denoted as S. Since the CWPF relies on a linear relationship between the relative yield and the relative evapotranspiration (from §2.4.2), S must achieve a value in the vicinity of 1 which reflects the ability of the CWPF to predict true values. The second characteristic entails a coefficient of determination, denoted as r^2. The coefficient of determination is regarded by some scientists as a goodness of fit measurement for estimated data points. This coefficient aims to partially measure how close data points are to the regression surface [5]. The third characteristic is the index of agreement of Wilmot [143], denoted as D. This index is regarded as a standardised measure for measuring the degree of model prediction error and usually varies between the values of 0 and 1. The desired index of agreement to achieve is 1. The fourth characteristic entails a *root mean square error* (RMSE) value. This value is known as a performance metric which indicates the average deviation between the measured and calculated values. The fifth characteristic is the *mean absolute error* (MAE) which is expressed as a percentage of the mean. This value is an indication of the average absolute error between the measured field values and the calculated values. Finally, the sixth characteristic entails a 80% accuracy frequency which computes the population of simulated values that correspond with 20% of the measured field values, and is usually presented as a percentage. The statistical results obtained by De Jager [21] for the CWPF (2.7)–(2.11) are summarised in Table 2.2. The reliability column indicates the statistical requirement that should be met by each CWPF in order to be considered as an accurate estimation of true yields [21].

From Table 2.2, it is clear that the statistical requirements were met for the majority of CWPFs, except for the Stewart and Hagan FS (2.9) and the model of Jensen (2.11). Both these models fall short of the $D80(\%)$ requirement, with the Stewart and Hagan FS (2.9) achieving a value of 72% and the the model of Jensen (2.11) achieving a value of 76%. The statistical results for r^2, D and MAE parameters were in close proximity for all the models. From the CWPFs, the Stewart multiplicative (2.8) obtained the most accurate estimation with the second lowest RMSE = 599 kg/ha, the highest $D80(\%) = 90\%$ value and a $S = 1.03$ value compared to the rest of the models. The Stewart additive (2.10) obtained the second most accurate estimation, with the lowest RMSE = 595 kg/ha value, a $D80(\%) = 86\%$ value and $S = 1.05$ value.

Statistical parameter	Crop water production functions (2.7)	(2.8)	(2.9)	(2.10)	(2.11)	Reliability criterion
S	0.97	1.03	1.07	1.05	1.02	0.9-1.1
r^2 (%)	90	91	91	93	92	>80
D	0.97	0.98	0.97	0.98	0.97	>0.8
MAE (%)	12	10	11	9	11	<20%
RMSE (kg.ha^{-1})	675	599	676	595	615	<700
D80 (%)	83	90	72	86	76	>80%

TABLE 2.2: *The statistical results obtained by De Jager [21] for CWPF (2.7)–(2.11) when estimating the final spring wheat yield.*

Furthermore, the De Jager additive (2.7) obtained the third most accurate estimation with a RMSE = 675 kg/ha, $D80(\%) = 83\%$ value and $S = 0.97$ value, followed by the Jensen exponential (2.11) which obtained a RMSE = 615 kg/ha, $D80(\%) = 76\%$ value and $S = 1.2$ value. The model of Stewart and Hagan FS (2.9) was considered the least accurate estimation since it achieved a low RMSE = 676 kg/ha, a low $D80(\%) = 72\%$ value and $S = 1.07$ value. De Jager [21] concluded that the models of Stewart multiplicative (2.8), Stewart additive (2.10), De Jager additive (2.7) and Jensen exponential (2.11) are proven satisfactory to be used for decision support purposes. It is not clear why the Jensen exponential (2.11) was classified satisfactory for the use of DSSs, however, it may be the case that the Jensen exponential model (2.11) showed exceptional results in the other validation characteristics. Finally, the Stewart and Hagan FS (2.9) was proven to be unsatisfactory for decision support purposes, and may be due to the fact that a constant β-parameter was used for the entire growth period instead of specific growth periods.

Igbadun *et al.* [66] also made a contribution towards evaluating four different CWPFs. The aim was to test the suitability, capability, accuracy and applicability of each CWPF. The four models that Igbadun *et al.* [66] considered included two multiplicative-type of models from Jensen [69] and Minhas *et al.* [86], and two additive-type of models from Stegman *et al.* [119] and Bras and Cordova [10]. The model from Stegman *et al.* [119] is merely a modified version of the model proposed by Stewart *et al.* [121], whereas the model from Bras and Cordova [10] was also subjected to a small modification. Igbadun *et al.* [66] describes the model of Jensen as of multiplicative-type, whereas De Jager [21] referred to the model as exponential-type. It is important to note that the multiplicative-type Jensen [69] and the exponential-type Jensen [69] are the same, but are referred to differently in the literature. Furthermore, Igbadun *et al.* [66] used five validation statistical characteristics that differ from the six characteristics used by De Jager [21]. The model of Jensen [69] may, therefore, be used as a linkage between the two statistical analysis such that an indication is given of the proximity where the tested models of De Jager [21] may be placed in the study done by Igbadun *et al.* [66].

In the study of Igbadun *et al.* [66], a field experiment was also conducted in order to determine the accuracy of each model. This experiment was conducted in the Mkoji sub-catchment of the Great Ruaha River basin which is located in Tanzania. The experiment adopted a *randomised complete block design*[7], which also enabled the implementation of multiple water irrigation strategies for the TMV1-ST variety of maize. The maize was planted in clay and clay loam soils which are indigenous to the Usangu plain in Tanzania. Furthermore, eight different treatments were applied to the crop, one of which the crop water requirements were maintained throughout the growth period, and some of the others where the weekly crop water requirements were not

[7]This design describes the standard field layout when conducting agricultural experiments whereby different treatments are randomly assigned to each block individually.

met. The irrigation demand for the first growth period was maintained for all eight different treatments to ensure that the crop is well developed before exposed to moisture stress. The experiment was conducted over two seasons. Maize was planted on the 24^{th} June for the 2004 season and on the 6^{th} July for the 2005 season. For the 2004 season, the field data was used to determine the moisture stress sensitivity indices (yield response factors) for each model, whereas the yield from the 2005 season was used to compare the simulated results with the predicted outcome.

The CWPF tested by Igbadun *et al.* [66] is also founded in the relationship in equation (2.6). The first model is the model proposed by Jensen [69], as previously described in equation (2.11). The second model is the model proposed by Minhas *et al.* [86], and makes use of a multiplicative-type of method to combine yield reductions. This model is expressed as

$$\frac{Y_x}{Y_m} = \prod_{g=1}^{G} \left(1 - \left(1 - \frac{ET_a}{ET_m}\right)_g^2\right)^{\eta_g}, \tag{2.12}$$

where η_g denotes the Minhas' moisture sensitivity index, and the other parameters are as previously defined in equation (2.7). The third model is the model proposed by Stegman *et al.* [119] which is of additive-type. This model is expressed as

$$\left(1 - \frac{Y_x}{Y_m}\right) = \sum_{g=1}^{G} \gamma_g \left(1 - \frac{ET_a}{ET_m}\right)_g, \tag{2.13}$$

where γ_g denotes Stewart's yield reduction coefficient, and the other parameters are as previously defined in equation (2.7). The fourth and final model is the model of Bras and Cordova [10] which uses an additive-type of method to combine the yield reductions. This model is expressed as

$$\frac{Y_x}{Y_m} = \sum_{g=1}^{G} \delta_g \left(\frac{ET_a}{ET_m}\right)_g, \tag{2.14}$$

where δ_g is denoted as the Bras and Cordova's moisture sensitivity index, and the other parameters are as previously defined in equation (2.7). The original Bras and Cordova model defined the denominator in the ratio of actual evapotranspiration to maximum potential evapotranspiration as the potential evapotranspiration, and the moisture stress sensitivity index was taken as a coefficient for each week of the irrigation cycle. In the modified equation (2.14), the moisture stress sensitivity index was defined for the entire growth period and the denominator of the evapotranspiration ratio was taken as the evapotranspiration from the non-stressed treatment for the entire growth period.

Igbadun *et al.* [66] made use of five statistical validation characteristics to determine the accuracy of each CWPF, similar to De Jager [21]. Of the five characteristics, the first includes the *average error of bias* (AE) which describes the average error between the estimated values and true values. The second entails the RMSE, which is similar to the study done by De Jager [21]. It is important to note that a different unit was used by Igbadun *et al.* [66]. The third characteristic includes the *coefficient of variation* (CV), which indicates the degree of precision (in %) between the estimated values and measured field data. The fourth characteristic entails the *modelling efficiency* (EF) and explains the degree to which the CWPFs are able to predict relative yields that best fit the field measured data. The fifth and final characteristic entails the *coefficient of*

residual mass (CRM), and indicates to what extend the prediction was over predicted or under predicted (in %). An over-prediction is indicated as a negative value whereas an under-prediction is indicated as a positive value [94]. A thorough description of the five statistical measures used by Igbadun *et al.* [66] may be found in the paper of Panda *et al.* [94]. The statistical results obtained by Igbadun *et al.* [66] for CWPFs (2.11)–(2.14) are summarised in Table 2.3.

Statistical parameter	Crop water production functions (2.11)	(2.12)	(2.13)	(2.14)
AE	0.10	0.17	0.13	0.08
CV (%)	17.52	28.21	21.32	16.01
RMSE	0.12	0.20	0.15	0.11
EF	0.97	0.93	0.96	0.98
CRM (%)	-15	-25	-18	-12

TABLE 2.3: *The statistical results obtained by Igbadun et al. [66] for CWPFs (2.11)–(2.14) when estimating the final maize yield.*

Igbadun *et al.* [66] concluded that the estimated yields were in close proximity to the measured field data. The Bras and Cordova additive (2.14) achieved the lowest AE with a value of 0.08, whereas the model of Jensen (2.11) achieved a value of 0.10, the Stegman additive (2.13) achieved a value of 0.13, and the Minhas multiplicative (2.12) achieved a value of 0.17, in a descending order of performance. Moreover, Igbadun *et al.* [66] considered that the CV obtained by each model is a fairly good performance, since $70\% - 80\%$ of the variations in the field measured data and estimated yields were captured. Here, the Bras and Cordova additive (2.14) achieved the best result with a CV value of 16.01% whereas the Minhas multiplicative (2.12) achieved the worst result with a value of 28.21%. For the RMSE, the Bras and Cordova additive (2.14) achieved the best result with a value of 0.11, whereas the model of Jensen (2.11) achieved the second best result with a value of 0.12, the Stegman additive (2.13) achieved the third best result with a value of 0.15, and the Minhas multiplicative (2.12) achieved the worst result with a value of 0.20. The overall modelling efficiency of the CWPFs were considered very high ($> 90\%$), implying that the capability of the models to predict relative yields are sufficient to best fit the measured field data. In contrast, the CRM indicates that all the models over-predicted the field-measured data by 12%–25%. Here, the Bras and Cordova additive (2.14) and the model of Jensen (2.11) achieved the best results with 12% and 15% respectively, whereas the Minhas multiplicative (2.12) achieved the worst result with 25%.

In conclusion, keeping in mind that empirical models rarely simulate field data perfectly due to a degree of inherent variability of field data, the performance of all the models were considered fairly adequate. Comparing the results from Table 2.3, it is clear that the model of Jensen (2.11) and the Bras and Cordova additive (2.14) performed better in comparison to the Minhas *et al.* multiplicative (2.12) and the Stegman additive (2.13). An important consideration that Igbadun *et al.* [66] highlighted was the fact that the performance of the models may be enhanced by calibrating the moisture stress indices for a specific area. In summary, the model of Jensen (2.11) and the Bras and Cordova additive (2.14) are recommended.

In the FAO Irrigation and Drainage paper No. 66 [118], the relationship between relative yield and relative evapotranspiration, as denoted in equation (2.6), was addressed, and resulted in the proposal of a simple equation, previously denoted as equation (2.3). The aforementioned relationship is fundamental in the yield response factor approach, and the procedure thereof was documented and published in the FAO Irrigation and Drainage Paper Nr. 33 [35], and was regarded as a milestone publication. Furthermore, this approach was widely used for a broad range of applications, and was made possible since the equation (2.3) may be applied to all agricultural

crops, and includes herbaceous crops, trees and vines [118]. Irregardless of the results obtained in the studies done by De Jager [21] and Igbadun *et al.* [66], the model proposed by Doorenbos and Kassam [35] in equation (2.3) and the model of Jensen [69] are regarded as the most widely used relationships in literature when estimating crop yield response in terms of water supply [54]. The experiments done by the aforementioned authors are influenced by the calibration of moisture stress indices for specific areas. This may imply that the results portrayed by the models may be biased in the sense that the specific area and the performance of the calibration for moisture stress indices may suite an additive-type of model better than a multiplicative-type of model, or vice versa. In an unbiased manner, the multiplicative-type Stewart *et al.* [121] CWPF may be used in conjunction with the relationship described by Doorenbos and Kassam [35] for DSS, as proposed by De Jager [21]. The combination forms a multiplicative-type of model, and is expressed as

$$Y_j^{actual} = Y_j^{average} \times \prod_{g=1}^{G} \left(1 - K_{jg} \left(1 - \frac{ET_a}{ET_m} \right)_{jg} \right), \quad \text{for } j \in J, \ g \in G, \tag{2.15}$$

where j denotes the respective crop type, and remainder parameters are as previously defined in equation (2.7). If it is possible to obtain location specific climatic data, location specific crop yield response factors, location specific crop coefficients and crop water requirements per location, equation (2.15) may be used for a number of crops that are identified in Table 2.4. From the paper "Crop yield response to water" by Steduto *et al.* [118], multiple yield response factors are also available which may be used in conjunction with equation (2.15).

Alfalfa	Grape	Potato	Sunflower
Banana	Groundnut	Safflower	Tobacco
Bean	Maize	Sorghum	Tomato
Cabbage	Onion	Soybean	Water melon
Citrus	Pea	Sugarbeet	Wheat (winter)
Cotton	Pepper	Sugarcane	Wheat (spring)

TABLE 2.4: *A number of crops which may be modelled using CWPFs for estimating final crop yield [35].*

2.4.5 The costs of crop production

In the farming industry, farmers have to manage multiple cost aspects associated with crop production. Within each individual farm environment, the associated cost may vary according to the climatic conditions and environmental conditions. To analyse the economic implications affected by irrigation strategies using CWPF, it is important to quantify costs that may be encountered in the farming industry. A brief overview is given of the different cost associated with crop production, as well as the categorisation of these costs.

There are two categories with respect to farming costs in the farming environment, and is known as the *overhead cost* and *variable cost*. The overhead cost (also known as fixed cost) is unrelated to the level of crop production, meaning that these expenses must be paid regardless of the crop production volume. Here, an increase in crop production results in the attenuate of overhead cost, and ultimately producing units at a cheaper cost. This principle is known as achieving economies of scale [88]. Furthermore, Makeham and Malcolm [81] divided overhead costs into two different categories, namely total overhead costs and operating overhead costs. The aforementioned refers to unavoidable costs that must be met, and includes living costs, worker wages,

rent, governmental taxes, *etc.* The latter, on the other hand, refers to the cost associated with annual business operation, and includes registration of vehicles, depreciation, imputed labour, *etc.* The variable cost, however, is directly related to the level of production, meaning that an increase in production ultimately results in an increase in costs. Here, achieving economies of scale will result in higher variable cost. A few examples of variable cost includes fertilisers, pesticides, machinery, harvesting, drying, marketing, *etc.* The above mentioned examples portray only some of the costs expected in crop production, while multiple other expenses may form part of the budget relating to each respective farming industry.

From the Computus management information system [19], enterprise budgets were formed for the production of crops and indicates the expected overhead costs and variable costs for a number of crops, respectively, as well as the expected production break-even point. From the South African Agriculture, forestry and fisheries department, multiple trends have also been documented for a number of crops [32]. Moreover, from the SAFEX website [58], the expected overhead costs and the variable costs with respect to specific areas may also be found. This information may be used to form opinions with regards to overhead costs and variable costs associated with crop production, and may be also be used to formulate realistic farming cost budgets. Farmers may also decide to what extent the overhead costs, variable costs and some additional costs may affect the estimated total profit from crop yield.

2.4.6 Crop production DSSs

The process of determining crop yield response factors and reference crop evapotranspiration using data from a field experiment can be complex, costly and time-consuming. CropWat 8.0 [50], however, is a computerised DSS that enables the computation of crop yield response factors, reference crop evapotranspiration, and many more parameters associated with irrigation practices. The cardinality of these parameters are proven essential when estimating the final crop yield using CWPF, as proven in §2.4.4.

CropWat 8.0 was developed by the Land and Water Development Division of the FAO as a practical tool to assist farmers in the decision-making process regarding irrigation. The aim of this software is to determine effective irrigation methods using standard calculations. CropWat 8.0 can perform a number of calculations, such as the calculation of ET_o, K_y, crop water requirements and more specifically, design and manage of irrigation schedules and requirements [83]. The calculation procedure is based on two methodologies presented by the FAO Irrigation and Drainage Papers, namely No. 24 "Crop water requirements" [36] and No. 33 "Yield response to water" [35]. Incorporated in the CropWat 8.0 software is the ClimWat 2.0 [49] software which is a climatic database that allows the estimation of ET_o using multiple climatic conditions gained from various weather stations worldwide. Furthermore, irrigation schedules are developed according to a soil-water balance by implementing various aspects from water supply and irrigation management conditions. CropWat 8.0 specifically requires the use of three input parameters, and in the case of irrigation scheduling, an extra two input parameters are required. As already known, CWPFs require a number of parameters in order to calculate crop final yield, of which two of these parameters are selectable from CropWat 8.0. These parameters include the K_y value(s) for specific growth periods and the $ET_c^{(dec)}$ value(s), denoted as the crop evapotranspiration taking place over a period of 10 days.

After installing ClimWat 2.0, a number of weather stations may be selected from a list of countries. Each weather station contains data required for the estimation of reference crop evapotranspiration based on a specific area, and includes all the parameters required by equation (2.2), as described in §2.4.2. The selected data is then exported by ClimWat and imported by

CropWat. In Figure 2.9, the number of weather stations to select from are indicated by the white dots across South Africa, Namibia, Botswana, Zimbabwe, and Mozambique. On the right hand side of Figure 2.9, the 73rd weather station located in Montagu was selected for illustrative purposes.

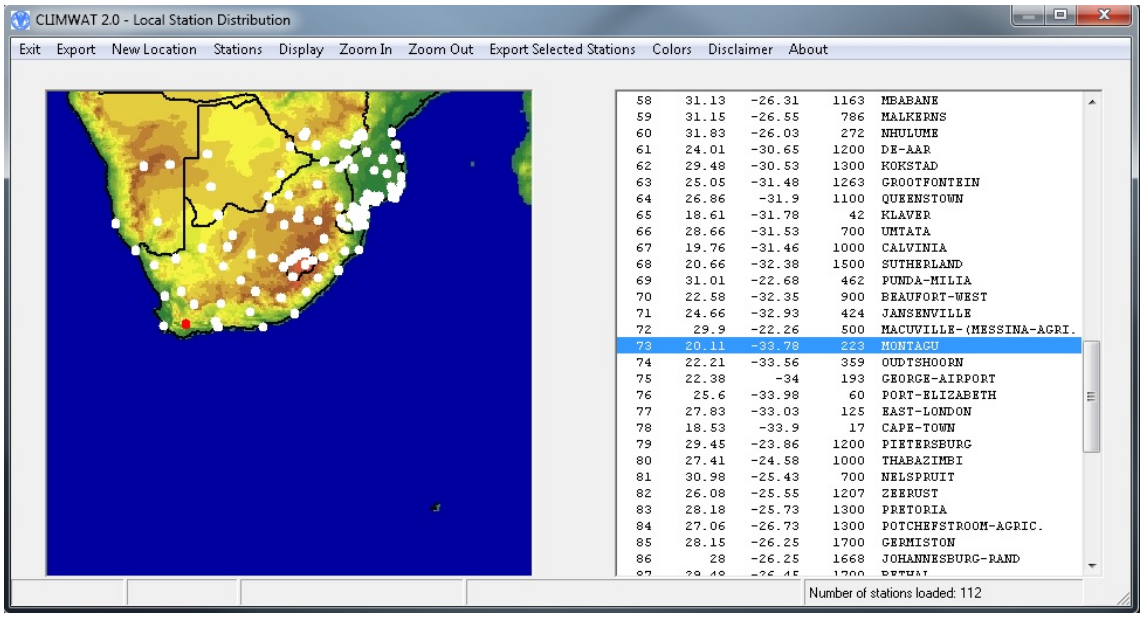


FIGURE 2.9: *A map of multiple weather stations scattered across Southern Africa that collect weather data for evapotranspiration purposes [49].*

After the ET_o is calculated, the response factor for each growth period may be determined. Potatoes were selected for the illustrative purposes of which the obtained results are illustrated in Figure 2.10 for each growth period respectively. The K_y values that correspond with each growth period are noted as $0.45, 0.80, 0.80, 0.30$, respectively, while the value 1.10 corresponds to the K_y of the full-season. Furthermore, the K_c value denotes the crop coefficient described in §2.4.2 and incorporates the *water stress coefficient*, which is denoted as K_s and describes the severity of the soil water deficit on the crop evapotranspiration, the Stage (days) which indicates the length of crop growth periods, and the critical depletion fraction, denoted as ρ, which indicates the soil moisture level where the first drought took place [116]. The last named parameters, however, are not necessary for the estimation of final yield using a CWPF, and therefore, only the K_y factor in conjunction with crop water requirements are required.

CropWat uses the K_y values, the ET_o to estimate the ET_c, and the ET_c values to estimate the crop water requirements for various selectable crops. In this case, the crop water requirements for potatoes are calculated by CropWat and illustrated in Figure 2.10. The crop water requirement calculations are executed for a period of 10 days, denoted as a decade, assuming that all months have 30 days for simplicity reasons [116]. The average daily crop evapotranspiration is calculated as $ET_c^{(day)} = K_c \times ET_o$, and the crop evapotranspiration per decade is calculated as $ET_c^{(dec)} = ET_c^{(day)} \times$ *the number of effective crop days*, which is normally taken as 10 days [116]. For the first and last decade, the number of effective crop days vary due to the fact that planting and harvesting dates do not coincide with the beginning and end of the month. This may also occur in any decade during the entire growth season. Furthermore, the effective rainfall (Eff rain as indicated in CropWat) indicates the expected effective volume of rainfall for a decade of days, and the irrigation requirement (Irr. Req. as indicated in CropWat) indicates the quantity of water that must be assigned to crops for irrigation when taking into account the average

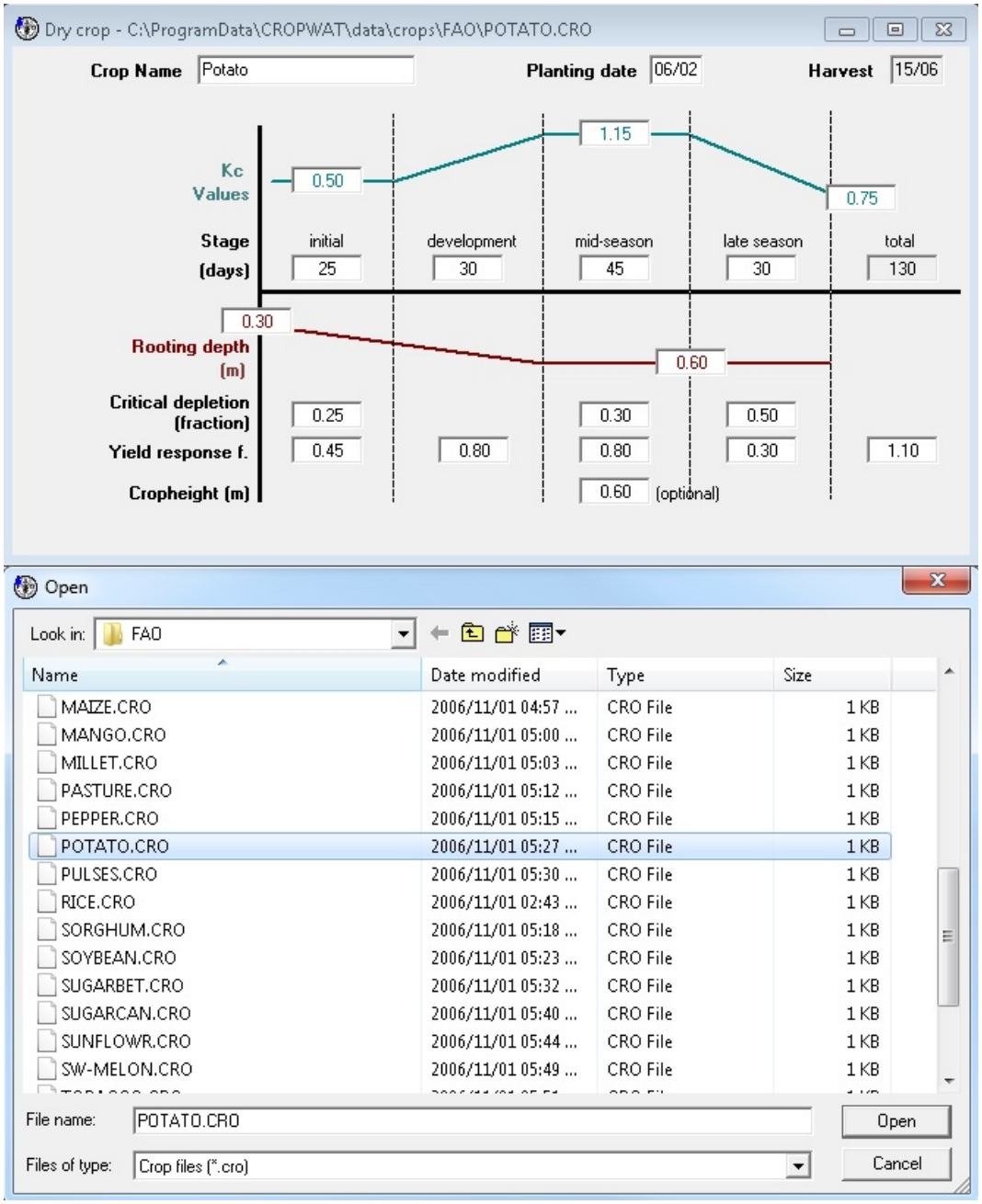


FIGURE 2.10: *A number of crops which are selectable in CropWat 8.0 and the corresponding parameters for the selected crop which may be used for irrigation purposes [50].*

historical rainfall [116]. Alternatively, the $ET_c^{(dec)}$ may be used to determine the crop water requirements for the entire growth season when the current farming practices are considered ineffective, and also when no rainfall occurs.

CropWat 8.0, however, is limited to a certain extent due to the significant variability of the yield response factor. Therefore, the AquaCrop [48] simulation model is developed by the land and water division of the FAO that is also known as a growth model for herbaceous crops and, is therefore considered as the follow up version of CropWat 8.0. AquaCrop, however, incorporates a different approach when analysing a number of factors affecting crop production and the environment. The original concepts related to the direct link between crop yield production and

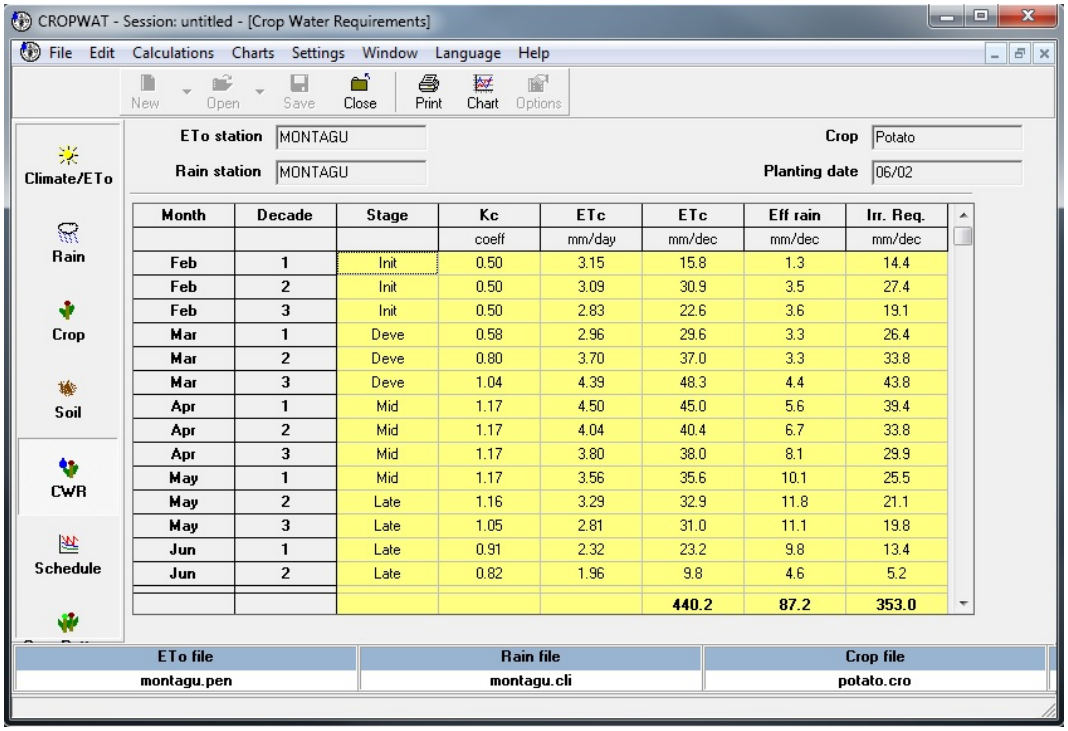


FIGURE 2.11: *The crop water requirements for potatoes calculated by using the weather data gathered from the Montagu weather station. The crop growth stage, crop coefficients, effective rain and irrigation requirement are also shown [50].*

water use, as proposed by Doorenbos and Kassam [36], is maintained in this model. However, due to the variability of the yield response factor, the soil evaporation is not considered whereas the productive crop transpiration and a biomass production is estimated from the actual crop transpiration using a water productivity parameter. The use of AquaCrop is not plausible due to the complexity of the new approach that is undertaken compared to the previous approach of Doorenbos and Kassam. The approach of AquaCrop may yield more accurate results, however, the implementation thereof is considered to be too complex when implementing optimisation in crop growth periods. Furthermore, the results obtained from AquaCrop may be used for validation purposes since there exists limited correspondence between the chosen approach and AquaCrop results. Placing careful emphasis on the accuracy of parameters when using the approach proposed by Doorenbos and Kassam [36], accurate estimations may be obtained which guide the decision-making process associated with irrigation purposes.

In the past 40 years, crop modelling at the hand of computing technologies has developed extensively, resulting in the formulation of a wide range of crop models, each with different levels of complexity and variation in the undertaken approach. Singels *et al.* [114] has dedicated a thorough study on multiple mathematical simulation models of the most famous crop growth models that contributed to managing crop production and developing of crop sciences in South Africa. A short overview of the seven major modelling initiatives in South Africa is given in the remainder of this section, according to Singels *et al.* [114].

ACRU

ACRU was introduced in the 1970s originating from a study by Shulze [111] on the catchment-based hydrological study in the Drakensberg of KwaZulu-Natal. The model has a multi-purpose

design that integrates multiple water budgeting components from the terrestrial hydrological system [114]. Furthermore, the application spectrum of this model stretches across a number of instances, namely crop yield modelling, designer hydrology, irrigation water supply and demand, surface runoff estimation, reservoir yield simulation, *etc.* Incorporated in this model is the utilisation of a multi-layer soil water budget that estimates the evapotranspiration, followed by empirical functions of evapotranspiration that estimates yields for a number of crops taking into account the crop development stage. However, this estimation is subjected to the availability of information from the location.

BEWAB

The abbreviation, BEWAB, originates from the Afrikaans term, "BEsproeingsWAterBestuursprogram", when translated to English means "the irrigation water management program", and was developed at the University of the Free State by the Department of Soil. The model serves as a daily DSS that assists farmers in the decision-making process with regards to the amount of water, as well as the timing of water applications. Incorporated in the model are CWPFs, non-linear crop water demand functions, and the locality of planting dates for each crop based on water use measurements. A universal transpiration efficiency theory was introduced by Strydom [123] and concepts were formulated by De Wit [25], Hanks [59] and Tanner and Sinclair [126] that resulted in consequently widening the application spectrum of this model. Now, modelling can be done on any irrigation scheme relative to the availability of local information, taking into account sprinkler and flood irrigation types [111].

CANEGRO

The CANEGRO model was developed in the response of questions from stakeholders in the SA suger industry, with the goal to limit the outbreak of the *Eldana saccharina* sugercane stalk borer. There was an initiative to reduce harvesting age from sugercane such that the damage from the *E. saccharina* is limited, since cutting sugercane older than 12 months proliferated the growth of these pests. Ultimately, the goal of CANEGRO was to direct and assist research in the Sugarcane industry, which later was given international recognition when the model was incorporated in the *DSSAT* [8] (Decision Support System for Agrotechnology Transfer) v3.1 and v4.5 packages. The application of this model, however, was limited to studies performed by scientist who had direct access to the code [111]. Some of the application avenues of the CANEGRO model resulted in crop forecasting, estimating potential and attainable yield, consultation studies such as climatic yield potential, *etc.*

CERES

The DSSAT model that was previously mentioned in the CANEGRO section, also includes the CERES-maize model. This model was initially adopted to fit South African weather conditions, although the main challenge that remained was the estimation of final yield over larger areas. For the Free State area, De Jager *et al.* [22] developed a maize yield modelling framework which later formed the basis of the current *National Crop Estimates System* (NCES) for maize. In the NCES system, the CERES-maize model was incorporated using South African input data

[8] DSSAT is classified as a support system that comprises out of 42 crop simulation models and tools that facilitate the effective use of these models. The establishment of such a model was motivated by a need for an integrated support system with regards to soil, climate, crops and management for more effective decision making.

and weather conditions. The application area of the CERES-model only focusses on the final yield estimation of maize, and enables farmers to compute the final yield for a range of natural occurrences, such as drought assessment and climate change [111].

PUTU

The PUTU models was introduced in 1968 by De Jager [23], and its name originates from the maize meal porridge, and evolved from the process called leaf photosynthesis. The goal of the PUTU system is to apply the fundamental laws of physics and chemistry and mathematical equations such that the growth and expansion processes subsumed in agronomical crops are thoroughly explained and quantified. The PUTU system has a wide range of applications. Some of which include the determination of irrigation requirements as well as crop water requirements, ENSO based drought monitoring, the quantification of production potential and risk for different wheat and maize production strategies, the optimisation of irrigation water, *etc.* [111].

SAPWAT

The development of the *South African Program Water* (SAPWAT) aimed to serve as a planning and management tool in the farming industry, enabling farmers to compute crop irrigation requirements. With the implementation of SAPWAT in conjunction with rainfall, evaporation and a water balance model, the crop evapotranspiration was assumed to be directly proportional to the American class A-pan evapotranspiration. The result of this was the development of irrigation research and technology, and ultimately leading to opportunities for improving potential crop evapotranspiration estimations. Included in the SAPWAT program was the general estimation of crop yield using the findings of Doorenbos and Kassam [36]. Today, SAPWAT is used to estimate irrigation water allocations, and includes connections linked to licensing and registering the use of agricultural water [111].

SWB

The *Soil Water Balance* (SWB) model was developed from a generic crop model that based its principles on a simple water balance model proposed by Campbell and Diaz [16]. In this model, the daily increment of biomass is simulated and subjected to a limited intercepted radiation or water supply. Furthermore, the partitioning of biomass to a number of crop components is effected by water, and also affecting crop canopies. The water soil balance is then simulated following a layered, cascading approach. The model was further developed due to practical needs in the irrigation agriculture. In this model, a number of crops are incorporated namely winter and summer vegetables, potatoes, field crops, *etc.* The application area of the SWB model stretches from feasibility studies using saline or neutralised aced-mine water for irrigating crops to predicting long-term environmental impacts and sustainability. This model has also proven as a very useful teaching aid and is implemented in various teaching organisations and companies [111].

2.5 Soil moisture management systems

The implementation of soil moisture management systems are becoming more common amongst farmers since these systems provide guidance in term of effective irrigation strategies. An imple-

mentation system such as this provides the ground work for a DSS, since it foresees information on irrigation intervals, irrigation quantities and meteorological data from the environment.

Soil moisture management systems consist of four components, namely weather stations, soil moisture probes, telemetry and computer software. A weather station is usually a compact device with meteorological sensors that aims to determine theoretic atmospheric demand, and when combined with soil moisture probes, a fine equilibrium between general evaporative demand and location specific soil moisture data may be achieved. The soil moisture probe is a shaft or column that is inserted into the ground, hosting one to ten moisture measuring sensors. This device uses electrical impedance technology to measure location specific soil moisture along the various sections of the probe column. In combination with a telemetry device and computer software, real-time data may be continuously logged by sending data to the telemetry device. This data is then passed to the computer and captured by the software [65]. A graphical illustration of a soil moisture probe is shown in Figure 2.12.

FIGURE 2.12: *A graphical illustration of a soil moisture probe, as developed by HydraWize [65], for the effective soil moisture management systems employed in the farming industry.*

As mentioned in §2.4.2, crop water requirements are governed by the evapotranspiration taking place, whereas the ratio between actual evapotranspiration and maximum evapotranspiration may be taken as the ratio of irrigation supply to irrigation demand. Since soil moisture probes measure the soil moisture content in real-time, the total evapotranspiration that takes place may be estimated by utilising this measurement technique. This procedure may be regarded as an even more accurate estimation of crop water requirements when compared to the computed crop water requirements by CropWat 8.0, as explained in §2.4.6. Real-time crop evapotranspiration that is location specific contributes towards the accuracy of estimating crop water requirement. Subsequently, a more accurate estimation of crop evapotranspiration may lead to may result in smaller error margins of the estimated total profit earned, yield production, as well as an irrigation schedule when incorporating the location specific crop evapotranspiration in a crop coefficient approach.

2.6 Limitations of the yield response factor approach

The procedure proposed by Doorenbos and Kassam [35] in the Irrigation and Drainage Paper No. 66 is a popular approach, and is therefore chosen by many economist and engineers. The relationship that describes the yield reduction relative to the reduction in evapotranspiration has been considered as a standard method in the development of planning models for water

allocation [118]. Till today, the approximation of final yield regarding herbaceous crops, vines and trees are based on the water requirements of crops.

The approach adopted by Doorenbos and Kassam [35] uses a single empirical derived yield response factor to predict crop yield in growth stages. This approach, however, is considered limiting since the accuracy of the empirical derived yield response factor affects the quality of the predicted crop final yield. A single-value, that is confined to a growth period, is used to predict the final yield and a small change in the margin of this value may result in large fluctuations of the estimated crop final yield. Furthermore, other factors such as, for example, nutrients, different cultivars, to name a few, may also affect the yield response to water and is not considered in the K_y factor. These factors may also vary according to geographical locations, climatic conditions and environmental conditions. Therefore, the accuracy of the K_y factor is essentially important in order to make accurate crop final yield predictions using CWPFs [118]. Moreover, according to the FAO No. 66, the calculated K_y value may be compared under a cooperative research programme which is carried out by the *International Atomic Energy Agency* [35].

The determination of K_y values was carried out by different studies for various crops under various different conditions. Doorenbos and Kassam [35] and FAO [52] proposed original values for K_y of which the results where compared under the cooperative research programme. Despite the robustness of the relationship proposed by Doorenbos and Kassam [35], no trend can be extracted from the variation in the K_y values proposed by Doorenbos and Kassam [35] and FAO [52] under various different conditions. Therefore, it may be concluded that the application of the approach by Doorenbos and Kassam [35] are sufficient for general planning, design of irrigation projects, and the rapid assessment of yield reductions under limited water supply [118].

2.7 Chapter summary

This chapter was devoted to a brief overview of the notions found in the literature with respect to agricultural prerequisites. This chapter opened in §2.1 with a brief introduction to water irrigation in the United States, and numerous investigations conducted on crops and crop water requirements during the period between 1890 and 1920. Next, in §2.2, the notions of evaporation and transpiration were described. These notions specifically plays an important role in crop water requirements. This was followed by a brief explanation on the method conducted in this thesis for computing evaporation from water surfaces in §2.3.

Next, in §2.4, a number of notions were comprehensively discussed as part of the method to compute final yield from crops. This entailed discussions on crop growth stages, the crop coefficiënt and the yield response factor, the maximum yield of crops, CWPFs, and crop production DSSs. Within this section, the CWPFs were statistically analysed in order to select a method that may predict crop final yield most accurately. This resulted in a method that combines the multiplicative-type Stewart *et al.* [121] CWPF and the crop-water production relationship proposed by Doorenbos and Kassam [35]. Moreover, in §2.5, a number of components were briefly described as part of soil moisture management systems which may be used to estimate crop water requirements more accurately. Next, in §2.6, the limitations associated with the adopted approach when estimating crop final yield were comprehensively discussed. This chapter closed in §2.7 with a summary of the chapter contents in this chapter.

CHAPTER 3

Multi-objective optimisation

Contents

This chapter is devoted to give an introduction on the fundamental concepts conforming to the field of multi-objective optimisation. The chapter opens in §3.1 with an introduction to computational complexity when solving optimisation problems, and is followed with a discussion on the general notions and concepts involved in multi-objective optimisation as well as a general formulation of *multi-objective optimisation problems* (MOOPs) in §3.2. These notions and concepts give rise to two spaces, namely a decision space and an objective space. Furthermore, the principles in the field of multi-objective optimisation that are discussed in this section merely serve as a broad exposition of the opening chapters in Deb's influential text, known as *Multi-objective optimisation using evolutionary algorithms* [26].

In §3.3, the convexity and non-convexity of a MOOP are thoroughly discussed. These notions are important since it may have an impact on the solution methodologies that may be used to solve such problems, and may also lead to shorter computational time when solving such problems. Next, in §3.4, the principle of dominance amongst solutions to a MOOP are introduced. In MOOPs, the aim is to find a set of solutions amongst a set of candidate solutions which are superior with respect to the other solutions in the set. In §3.5 the focus shift towards a discussion on a number of methods that are able to uncover the set of nondominated solutions in a set of candidate solutions. These methods include the naive and slow method, continuous update

approach and the method of Kung *et al.* [75]. The implementation of such methods positively contributes towards computational expense when solving large sets of candidate solutions.

Moreover, the well-known method of nondominated sorting is discussed in §3.6. This method is particularly important when candidate solutions need to be partitioned into different classes relative to their degree of dominance, as required by a number of algorithms in the literature. Finally, a comprehensive discussion on the weighted sum of objectives method is given in §3.7. This section focuses on the scalarisation of objective functions when solving multi-objective optimisation problems as single-objective optimisation problems. The chapter finally closes in §3.8 with a brief summary of the chapter contents.

3.1 Defining computational complexity

Computational complexity theory forms part of theoretical computer science where computational problems are grouped into classes relative to the resources required to solve them [70]. Multiple examples of these resources includes storage space, computational time, number of processes, and random bits of which the most widely recognised are time and space.

According to Henning and Van Vuuren [60], an *algorithm* may be defined as a procedural operation that is ordered in a sequence in which the goal is to solve mathematical problems within a finite number of steps. Algorithms are widely implemented in various industries to solve different kinds of computational problems, and in most cases, the efficiency of an algorithm is analysed in terms of its speed and required computer memory when executing tasks. This is important because for a given computation problem of a respective size, an algorithm must be able to solve the problem within a realistic time window using a computer with sufficient memory capacity.

When referring to *algorithmic complexity*, there are two variables of an algorithm to consider namely the *time complexity* denoted by $T_c(a)$, and the *space complexity* denoted by $S_c(a)$, where the magnitude of the input to the algorithm is denoted as a and the size of the problem is denoted as c. The time complexity variable represents the total amount of time required by a computer to execute an algorithm whereas the space complexity variable represents the total amount of space (*i.e.* memory) required for the same procedure. For a given input size of a problem, denoted by a, the *worst-case* complexity of an algorithm is assumed to be the largest values for $T_c(a)$ and $S_c(a)$. Since it is difficult to estimate the exact required resources to execute an algorithm, so-called *asymptotic upper bounds* are implemented which aim to seek upper bounds for $T_c(a)$ and $S_c(a)$. These bounds describe the change in functions $T_c(a)$ and $S_c(a)$ as $a \to \infty$. In order to describe the latter notion, let $g(a) = \mathcal{O}(h(a))$ where constants of $b \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ exist such that $0 \leq g(a) \leq bh(a)$ for all $a \geq n_0$ [60]. If the latter case is true, the function $g(a)$ is an asymptotic upper bound of the function $h(a)$ as $a \to \infty$, and it is said that function $g(a)$ is of *the order of* the function $h(a)$.

Considering *polynomial-time* algorithms, the time complexity is denoted as $\mathcal{O}(a^m)$ where a is defined as the size of the input instance to the algorithm, and $m \in \mathbb{R}^+$. *Decision theory* is the part of complexity theory which focuses on binary output to problems, and is interpreted as boolean values that either takes on a value `true` or `false` [139, 60]. The class of decision problems which is solvable by polynomial-time algorithms is denoted as **P** (*Polynomial*) whereas the **NP** (*Non-deterministic Polynomial*) class of problems contains all the decision problems for which polynomial-time algorithms answers `true`, given additional information related to the problem instance, known as a *certificate*. Moreover, the **co-NP** class of problems incorporates all the decision problems of which polynomial-time algorithms answers `false`, given additional

information related to the problem instance (again called a certificate) [70]. It is, however, sometimes difficult to find certificates for decision problems of the classes **NP** and **co-NP** although such certificates exist. A graphical illustration of the aforementioned complexity classes of decision problems is illustrated in Figure 3.1.

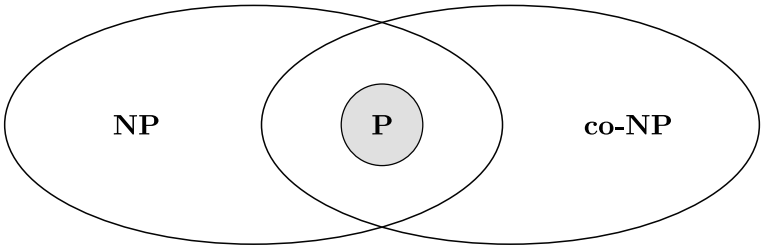


FIGURE 3.1: *The **NP**, **P** and **co-NP** complexity classes of decision problems [139].*

In order to determine whether a decision problem is at least as hard to solve as another, the notion of *reducibility* is introduced. Suppose B_1 and B_2 denote two decision problems of which B_1 is *polynomial-time reducible* to B_2 given that an algorithm A_1 exists that is capable of solving all instances of B_1, and also contains an algorithm A_2 as subroutine which is capable of solving all instances of B_2 such that A_1 is a polynomial time algorithm if A_2 is a polynomial time algorithm [60, 70].

If decision problem B_1 is polynomial-time reducible to B for all $B_1 \in \mathbf{NP}$, then decision problem B may be classified as a **NP**-hard problem. If decision problem $B \in \mathbf{NP}$, and B is considered to be **NP**-hard, the decision problem B may then be classified as **NP**-complete [139, 60, 70]. Since the class of **NP**-complete is a subset of the class **NP**, these problems are considered the most restrictive. The latter is also considered the most difficult class of decision problems to solve since solving such problems are computationally at least as hard to solve as any other decision problem in the class of **NP**. Considering the **NP**-complete and **NP**-hard computational classes, a classification scheme is formulated which contains all the classes when solving computational problems, and is illustrated in Figure 3.2.

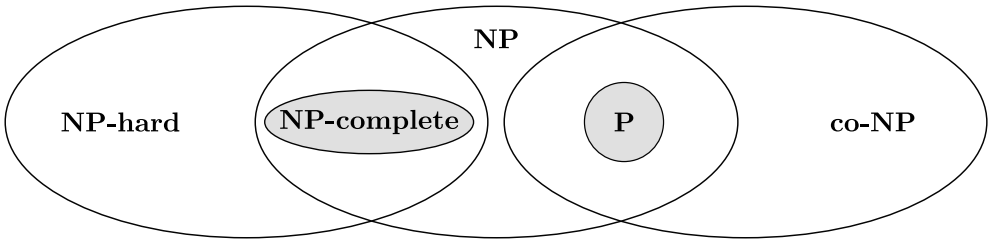


FIGURE 3.2: *The **NP**-hard, **NP**-complete, **NP**, **P** and **co-NP** complexity classes of decision problems [139].*

A solution to a computational problem typically consists out of real numbers (or a collection of real numbers) rather than binary values. As a result of this, decision problems of such nature may be considered as special cases to computational problems. Often computation problems may be solved in an efficient manner by incorporating algorithmic procedures and then repeatedly solving their associated decision problems. In this case, the notion of reducibility may also be employed to determine whether one *computational* problem is at least as hard to solve as another.

Suppose E_1 and E_2 denotes two computational problems. It may then be said that computational problem E_1 is polynomial reducible to the computational problem E_2 if an algorithm A_3

exists that is able to solve E_1, and uses an algorithm A_4 as a subroutine to solve computational problem E_2. Also, algorithm A_3 must run in polynomial time if algorithm A_4 runs in polynomial time when solving computation problem E_2 [70]. This yields that computational problem E_2 is at least as difficult as computational problem E_1 (computational problem E_1 is no harder when solving than computational problem E_2) if E_1 is polynomial time reducible to E_2. From Figure 3.2, the notion of **NP**-hardness and polynomial-time reducibility may also be generalised in such a manner that computational problems are accommodated in an obvious manner [60].

3.2 The general format of multi-objective optimisation problems

An optimisation problem may be defined as a problem in which the aim is to find the best possible solution (either an optimal solution or in close approximation of the optimal solution) to a given problem instance limited by a set of constraints [9, 17, 26]. For a *single-objective optimisation problem* (SOOP), the aim is to find a single solution that either maximises or minimises an objective function related to such a problem. In real-life scenarios, important decisions typically do not involve solving a single objective as a decision criterion, but rather involves a number of objectives which has to be solved simultaneously. One such example is the production of cars where the objectives are typically to produce good quality cars at a low production cost. These two objectives are, however, conflicting in nature since producing a good quality car typically results in a higher cost than producing a poor quality car. Such a problem is known as a MOOP where multiple objectives are to be maximised or minimised simultaneously in such a way so as to attain a number of good quality solutions that achieves compromises between objective function values [9, 17].

The general formulation of a MOOP is subjected to the following denotations. Let $f(\boldsymbol{x})$ denote a solution vector containing k number of discrete *decision variables*, $\boldsymbol{x} = [x_1, ..., x_k]$, to a MOOP. Moreover, let $\mathcal{D}$ denote the set of all approximate solutions to the MOOP (called the *decision space*) which is subjected to $\mathcal{P}$ objective functions. Furthermore, suppose there are two different types of constraints *i.e. equality* constraints and *inequality* constraints. Let A denote the number of equality constraints and let B denote the number of inequality constraints. The objective in a general MOOP is then

$$\text{maximise/minimise } \boldsymbol{f}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_{\mathcal{P}}(\boldsymbol{x})\}, \tag{3.1}$$

$$\text{subject to} \quad g_a(\boldsymbol{x}) = 0, \qquad a = 1, ..., A, \tag{3.2}$$

$$h_b(\boldsymbol{x}) \geq 0, \qquad b = 1, ..., B, \tag{3.3}$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \qquad i = 1, ..., k, \tag{3.4}$$

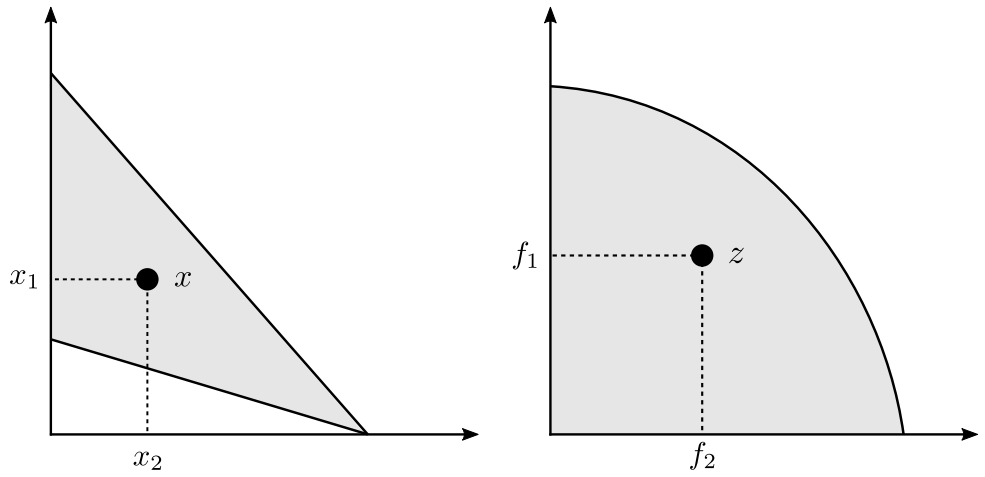
where $x_i^{(L)}$ and $x_i^{(U)}$ are constant lower and upper bounds for the decision variable x_i. Moreover, constraint set (3.2) represents the equality constraints and constraint set (3.3) represents the inequality constraints. The solution vector $\boldsymbol{x}$ is considered *feasible* if all the constraints in (3.2)–(3.4) are satisfied, and is considered *infeasible*[1] if at least one of the constraints in (3.2)–(3.4) is violated. The set of all feasible solutions to the MOOP constitutes the *feasible region* $\mathcal{F}$ (or the decision space) to the problem, and therefore $\mathcal{F} \subseteq \mathcal{D}$.

Each objective function in (3.1) may either be maximised or minimised, respectively. Many multi-objective solution methodologies, however, demand that all the objective functions of the

[1] Note that the entire decision space $\mathcal{D}$ need not be feasible.

problem either be maximised or minimised. In such a case, the *principle of duality* may be employed in order to transform an objective from a minimisation problem to a maximisation problem, or *vice versa*. This may simply be achieved by multiplying each objective function which have to be transformed with -1 [26].

A MOOP containing $\mathcal{P}$ objective functions also involves a $\mathcal{P}$-dimensional space called the *objective space*, and is denoted by $\mathcal{Z}$. For each solution vector $\boldsymbol{x}$ in the decision space $\mathcal{D}$, there exists a point $\boldsymbol{z}$ in the objective space — a k-dimensional solution vector in decision space is therefore mapped into a $\mathcal{P}$-dimensional vector in the objective space. This mapping is illustrated graphically in Figure 3.3 for the case where $k = 2$ and $\mathcal{P} = 2$.



(a) Decision space (b) Objective space

FIGURE 3.3: *A two-dimensional decision space of a bi-objective problem containing two decision variables illustrated in (a) and its corresponding two-dimensional objective space illustrated in (b). The solution vector $\boldsymbol{x}$ in the decision space and the mapping thereof to a point $\boldsymbol{z}$ in the decision space is also illustrated [26].*

3.3 Convexity and non-convexity in multi-objective optimisation problems

The notions of convexity and non-convexity are important aspects to consider when finding methods that are able to solve MOOPs. For example, many solution methodologies are easy to implement and is able to generate good quality solutions, however, are frequently ineffective when solving convex functions [26]. The definition of convex functions is as follows: A function f is *convex* if a line segment joining any pair of points in $\mathcal{H}$ is wholly contained in $\mathcal{H}$, given that the set of points is a subset of $\mathbb{R}^k$ [144]. Moreover, a function $f : \mathbb{R}^k \to \mathbb{R}$ is also considered *convex* if the inequality for any pair of vectors $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathbb{R}^k$ in

$$f\left(\lambda \boldsymbol{x}^{(1)} + (1-\lambda)\boldsymbol{x}^{(2)}\right) \leq \lambda f(\boldsymbol{x}^{(1)}) + (1-\lambda) f(\boldsymbol{x}^{(2)}) \tag{3.5}$$

holds for all $0 \leq \lambda \leq 1$. In order for the function f to be *nonconvex*, the inequality sign $\leq$ may simply be replaced with $>$ in (3.5) [144]. It is, however, possible that a function f can be neither convex nor nonconvex. A convex function joined by a line segment, as described earlier in this section, is graphically illustrated in Figure 3.4 for $\mathcal{P} = 1$ objective functions and $k = 1$ decision variables.

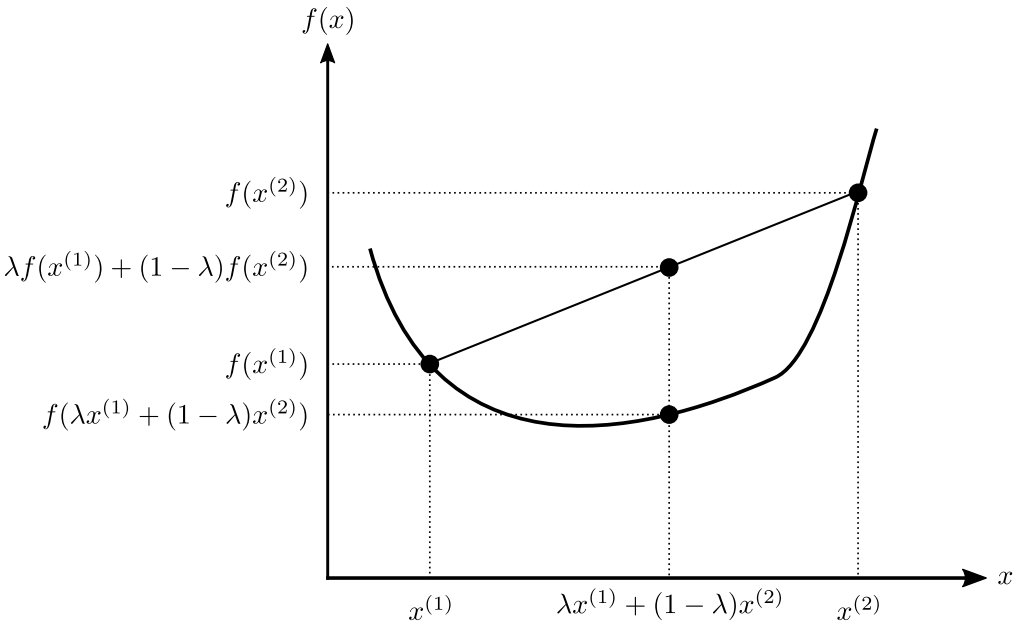


FIGURE 3.4: *A convex function with $\mathcal{P} = 1$ objective functions and $k = 1$ decision variables [26].*

There are three distinct properties associated with a convex function $f(\boldsymbol{x})$ [26], and include the following: (1) a linear approximation of $f(\boldsymbol{x})$ *underestimates* the value of $f(\boldsymbol{x})$ at any point along a straight line in $\mathbb{R}^k$ joining $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$, (2) the Hessian[2] of $f(\boldsymbol{x})$ is *positive definite*[3] for all $\boldsymbol{x} \in \mathbb{R}^k$, and (3) the local minimum of $f(\boldsymbol{x})$ is also a global minimum of $f(\boldsymbol{x})$.

In order to determine whether a function f is convex within a given region, the Hessian matrix of that function may be estimated, denoted by $\nabla^2 f$. This matrix should be checked for positive-definiteness for all the values of $\boldsymbol{x}$ in the region [26]. In order for the matrix to have positive-definiteness, all the principle minors of the Hessian must be non-negative for all the values of $\boldsymbol{x}$, and therefore, may then be classified as a convex function [144]. An alternative approach for testing for positive-definiteness is to determine the eigenvalues of the Hessian matrix. This yields that if all the eigenvalues of the Hessian is positive, then the matrix may be classified as positive-definite [26]. In a similar fashion, the function $f(\boldsymbol{x})$ is classified as non-convex if all the principle minors of the negation of the Hessian matrix $(-\nabla^2 f)$ are non-negative for all the values of $\boldsymbol{x}$ in the region. Ultimately, it may be shown that if a function $f(\boldsymbol{x})$ is non-convex, then all the vectors $\boldsymbol{x} \in \mathbb{R}^k$ and satisfy $f(\boldsymbol{x}) \geq 0$ collectively forms a convex set in $\mathbb{R}^k$. A set of non-convex constraints contribute to the feasible region of a MOOP, and result in the feasible region to embody a convex shape due to this.

Furthermore, a MOOP may be classified as convex if all the objective functions and the set of feasible solutions to the problem in (3.1)–(3.4) are convex — this is true if all the inequality constraints in (3.3) are non-convex and all the equality constraints in (3.2) are linear [26]. In conclusion, it is important to consider the convexity and non-convexity of MOOPs since well suited solution methodologies are required to solve MOOPs with distinctive characteristics.

[2]The Hessian denotes a square matrix estimated from all the second order partial derivatives for a function, and describes its curvature [144].

[3]Positive definiteness of a function refers to a symmetric matrix of which all its eigenvalues are positive.

3.4 Solution dominance in multi-objective optimisation problems

When comparing solutions to a MOOP with each another in objective space, the aim is to find a set of solutions which are superior to the remaining solutions with respect to all the objective functions given in the MOOP [9]. Such a set of solutions is known as the *nondominated* set of solutions, and the nature thereof is based on the notion of *dominance*. In order to show that solution $\boldsymbol{x}^{(1)}$ dominates solution $\boldsymbol{x}^{(2)}$, the *dominance relation* denoted by $\preceq$ is used and the dominance between two solutions is denoted by $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$. Therefore, if a solution $\boldsymbol{x}^{(1)}$ is said to dominate another solution $\boldsymbol{x}^{(2)}$ (*i.e.* $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$ and both solutions are candidate solutions to the MOOP (3.1)–(3.4), the following two conditions have to be satisfied. The first condition is that solution $\boldsymbol{x}^{(1)}$ is no worse than solution $\boldsymbol{x}^{(2)}$ in all $\mathcal{P}$ objective functions, and the second condition is that solution $\boldsymbol{x}^{(1)}$ is strictly better than solution $\boldsymbol{x}^{(2)}$ in at least one of the $\mathcal{P}$ objective functions [26, 84, 148]. If, however, any one of the two conditions is violated, it is said that solution $\boldsymbol{x}^{(1)}$ does not dominate solution $\boldsymbol{x}^{(2)}$, and it is denoted as $\boldsymbol{x}^{(1)} \npreceq \boldsymbol{x}^{(2)}$.

In order to illustrate the notion of dominance (and the use of the dominance relation), consider a bi-objective optimisation problem in which both objective functions f_1 and f_2 have to be minimised. Five candidate solutions to the problem are considered of which the objective function values for each candidate solution are illustrated in Figure 3.5.

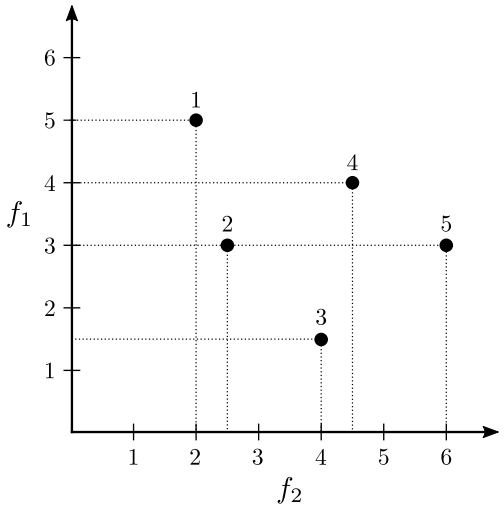


FIGURE 3.5: *Five candidate solutions in objective space for a bi-objective problem with objective functions f_1 and f_2. Both the objective functions f_1 and f_2 are to be minimised.*

Considering the five candidate solutions in Figure 3.5, it is difficult to find a single solution that is superior to all other solutions with respect to both objective functions. The notion of solution dominance may therefore be applied here in order to determine a single solution between a pair that performs better in the objective space. As an example, consider solutions 2 and 4. Solution 2 achieves a value of 3 in objective function f_1 and a value of 2.5 in objective function f_2. Solution 4, on the other hand, achieves a value of 4 in objective function f_1 and a value of 4.5 in objective function f_2. Solution 2 is therefore considered better than solution 4 since solution 2 achieves lower objective function values for both the objective functions f_1 and f_2. Considering the aforementioned conditions of solution dominance, both the conditions are thus satisfied which concludes that solution 2 dominates solution 4 (*i.e.* solution 2 $\preceq$ solution 4), or in other words, solution 2 is superior to solution 4.

Now consider solutions 1 and 3. Solution 1 achieves a value of 5 in objective function f_1 and a value of 2 in objective function f_2, while solution 3 achieves a value of 1.5 in objective function

f_1 and a value of 4 in objective function f_2. With respect to objective function f_1, solution 3 is better than solution 1, however, solution 1 is better than solution 3 in objective function f_2. Neither of the aforementioned conditions for solution dominance are therefore satisfied which concludes that neither of these two solutions dominate one another (no solution is superior to the other).

From the prior examples of solution dominance, it is clear that the dominance relation provides a way to compare candidate solutions to MOOPs for superiority. Therefore, the dominance relation is used in various multi-objective solution methodologies for finding the nondominated set of solutions to MOOPs of the form (3.1)–(3.4) [18, 26].

3.4.1 Properties of solution dominance

The comparisons of solutions among a set of candidate solutions to determine the nondominated set give rise to distinct properties associated with solution dominance in MOOPs. When comparing a solution $\boldsymbol{x}^{(1)}$ with another solution $\boldsymbol{x}^{(2)}$ using the dominance relation described in the previous section, three possible outcomes may occur [26]. These outcomes are as follows: (1) solution $\boldsymbol{x}^{(1)}$ dominates solution $\boldsymbol{x}^{(2)}$ (*i.e.* $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$), (2) solution $\boldsymbol{x}^{(2)}$ dominates solution $\boldsymbol{x}^{(1)}$ (*i.e.* $\boldsymbol{x}^{(2)} \preceq \boldsymbol{x}^{(1)}$), or (3) neither solution $\boldsymbol{x}^{(1)}$ dominates solution $\boldsymbol{x}^{(2)}$ nor does solution $\boldsymbol{x}^{(2)}$ dominates solution $\boldsymbol{x}^{(1)}$ (*i.e.* $\boldsymbol{x}^{(1)} \npreceq \boldsymbol{x}^{(2)}$ nor $\boldsymbol{x}^{(2)} \npreceq \boldsymbol{x}^{(1)}$). These outcomes give rise to three distinct characteristics associated with the dominance relation, namely:

Asymmetry: If solution $\boldsymbol{x}^{(1)}$ dominates solution $\boldsymbol{x}^{(2)}$ (*i.e.* $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$), solution $\boldsymbol{x}^{(2)}$ cannot dominate solution $\boldsymbol{x}^{(1)}$ (*i.e.* $\boldsymbol{x}^{(2)} \npreceq \boldsymbol{x}^{(1)}$). This implies that the dominance relation is *asymmetric*.

Non-reflexicity: Solution $\boldsymbol{x}^{(1)}$ cannot dominate itself according to the definition of dominance. This implies that $\boldsymbol{x}^{(1)} \npreceq \boldsymbol{x}^{(1)}$ and the dominance relation is *non-reflective*.

Transitivity: If solution $\boldsymbol{x}^{(1)}$ dominates solution $\boldsymbol{x}^{(2)}$ (*i.e.* $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(2)}$), and solution $\boldsymbol{x}^{(2)}$ dominates solution $\boldsymbol{x}^{(3)}$ (*i.e.* $\boldsymbol{x}^{(2)} \preceq \boldsymbol{x}^{(3)}$), then solution $\boldsymbol{x}^{(1)}$ dominates solution $\boldsymbol{x}^{(3)}$ (*i.e.* $\boldsymbol{x}^{(1)} \preceq \boldsymbol{x}^{(3)}$). This implies that the dominance relation is *transitive*.

Finally, it should also be noted that when solution $\boldsymbol{x}^{(1)}$ does not dominate solution $\boldsymbol{x}^{(2)}$, it does not imply that solution $\boldsymbol{x}^{(2)}$ dominates solution $\boldsymbol{x}^{(1)}$.

3.4.2 The view of Pareto optimality

Consider again the five candidate solutions in Figure 3.5 from §3.4. Comparing these solutions in objective space using the dominance relation, it is now easy to compute the dominating set of solutions. The nondominated set of solutions $\{1, 2, 3\}$ therefore contains the solutions in which no solution in that set dominates one another. In addition, these three solutions dominate all the other candidate solutions (*i.e.* solutions 4 and 5), and are therefore called the *nondominated set of solutions* to the MOOP. This yields that the set of solutions are not *inferior* (or superior) to other solutions in the nondominating set and is *superior* to the remaining candidate solutions [18, 26]. If $\mathcal{G}$ denotes the finite set of candidate solutions to a MOOP, the nondominated set of solutions $\mathcal{G}'$, which is a subset of $\mathcal{G}$, are those solutions which are not dominated by any other member in $\mathcal{G}$. Moreover, the resulting nondominating set of solutions $\mathcal{G}'$ is also known as the *Pareto optimal* set of solutions to a MOOP, and the result thereof is a *Pareto front* when plotted in objective space. The Pareto front typically entails a set of solutions on a hyper-edge

of the feasible region, and the orientation thereof depends on the type of objective functions in the MOOP which are either to be maximised and/or minimised. The set of nondominated solutions that are found for the entire feasible region is called the *globally Pareto optimal* set of solutions [18, 26].

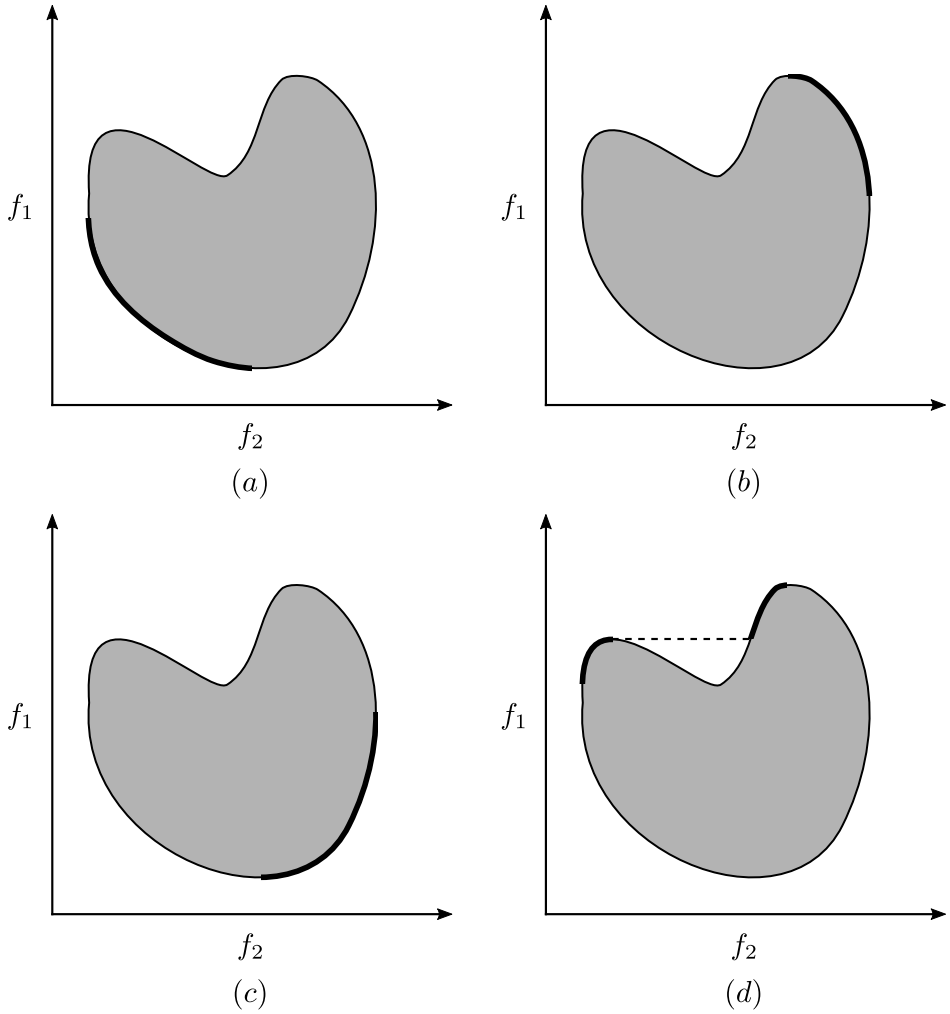
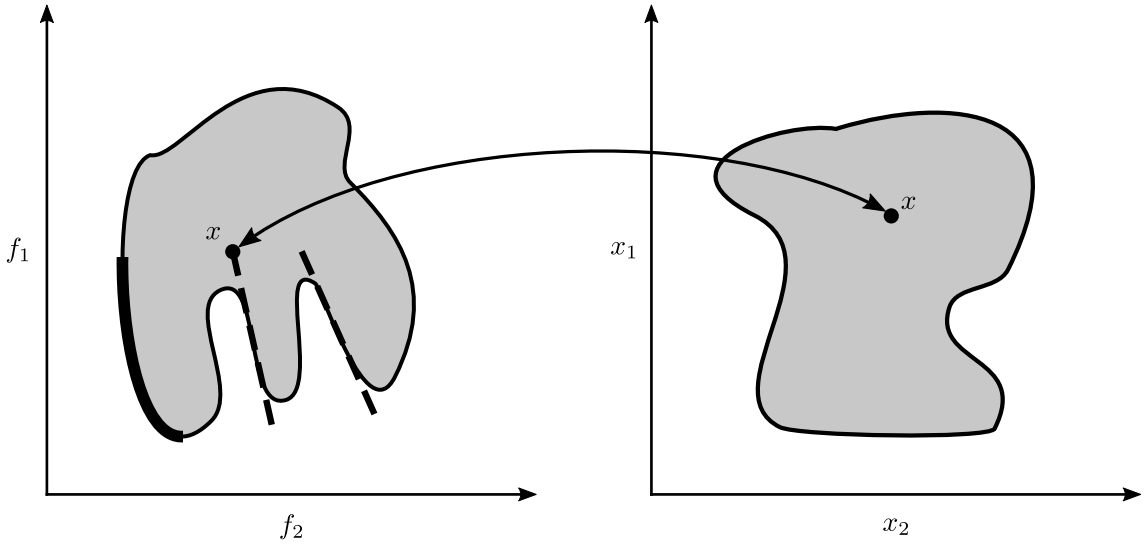


FIGURE 3.6: *Globally Pareto optimal curves for four different objectives of two objective functions f_1 and f_2. In (a) the aim is to minimise both the objective functions f_1 and f_2, and in (b) the aim is to maximise both the objective functions f_1 and f_2. Moreover, in (c) objective function f_1 is minimised and objective function f_2 is maximised while in (d) objective function f_1 is maximised and objective function f_2 is minimised. The variety of fronts is adopted from [26].*

There are four possible scenarios with regards to the orientation of the Pareto front subjected to a bi-objective optimisation problem. The four possible orientations are illustrated graphically in Figure 3.6. The solid dark curve presents the globally Pareto front for the problem, while the shaded area presents the feasible region to the MOOP in objective space for the same problem. Figure 3.6 (a) portrays a Pareto front in which both the objective functions f_1 and f_2 are to be minimised, while Figure 3.6 (b) portrays a Pareto front in which both the objective functions f_1 and f_2 are to be maximised. If, however, the objective function f_1 is to be minimised and objective function f_2 is to be maximised, the Pareto front, as portrayed in Figure 3.6 (c), is obtained. Note that due to the shape of the feasible region, the Pareto front is a union of two disconnected Pareto fronts. Finally, if objective function f_1 is to be maximised and objective function f_2 is to be minimised, the Pareto front illustrated in Figure 3.6 (d) is obtained. The

Pareto fronts obtained for a number of combinations with regards to the maximisation and/or minimisation of objective functions illustrated in Figure 3.6 are computed for the same feasible region.

Similar to SOOPs, locally Pareto optimal solutions may also be found in MOOPs. The definition of a local set of Pareto optimal solutions is that if no solution $\boldsymbol{y}$ exist to a MOOP that dominates any member of $\boldsymbol{x} \in \mathcal{G}$ in the neighbourhood of $\boldsymbol{x}$ such that $||\boldsymbol{y} - \boldsymbol{x}|| < \epsilon$, where ϵ is a small positive number, then the solutions in $\mathcal{G}$ forms a locally Pareto optimal set of solutions to a MOOP [27, 85]. This implies that a globally Pareto optimal set of solutions may constitute a locally Pareto optimal set of solutions, however, the converse is not necessarily true. In Figure 3.7 (a), an example of a globally Pareto front as well as two locally Pareto fronts are illustrated. The global Pareto front is represented by the dark solid curve while the two locally Pareto fronts are presented by the dark dotted curves. Furthermore, the phenomenon whereby no solution can be found that dominates any other member in the locally Pareto optimal set of solutions when perturbed in the solution space is also graphically illustrated in Figure 3.7 (b).



(a) Objective space (b) Decision space

FIGURE 3.7: *A globally Pareto front, represented by a dark solid curve, and two locally Pareto fronts, represented by two dotted curves, are illustrated in (a). These Pareto fronts are the result of a bi-objective optimisation problem where both the objective functions f_1 and f_2 are minimised [26]. When a solution of a locally set of Pareto optimal solutions is perturbed in the solutions space, a solution corresponding in the decision space is found, as illustrated in (b).*

3.4.3 Strong dominance and weak Pareto optimality

The notion of dominance naturally gives rise to the notion of strong or weak Pareto optimality among solutions in a candidate set of solutions. In general, the dominance relation is associated with *weak* dominance between solutions. The definition of strong dominance is as follows: solution $\boldsymbol{x}^{(1)}$ strongly dominates solution $\boldsymbol{x}^{(2)}$ if solution $\boldsymbol{x}^{(1)}$ strictly performs better than solution $\boldsymbol{x}^{(2)}$ in all $\mathcal{P}$ objective functions of a MOOP of (3.1)–(3.4) [115]. In order to illustrate the notion of strong dominance, consider again the five candidate solutions in Figure 3.5. It is clear that solution 3 strongly dominates solution 4, and that solution 2 does not strongly dominate solution 5 (weak dominance) since a similar objective function value is obtained for objective f_1 between solutions 2 and 5. It may therefore be said that if a solution $\boldsymbol{x}^{(2)}$ strongly dominates solution $\boldsymbol{x}^{(4)}$, then $\boldsymbol{x}^{(2)}$ also dominates $\boldsymbol{x}^{(4)}$ weakly, but not necessarily *vice versa*.

The notion of strong dominance may also be used to define a *weakly nondominated set*. For example, let $\mathcal{G}$ denote a set of candidate solutions to a MOOP and let $\mathcal{G}'$ denote a set of weakly nondominated solutions. Therefore, a weakly nondominated set $\mathcal{G}' \subseteq \mathcal{G}$ are those solutions that are not strongly dominated by any other member in $\mathcal{G}$ [18, 115]. The cardinality of a weakly nondominated set of solutions is therefore either greater or equal to the cardinality of a nondominated set of solutions, as obtained through the means of the dominance relation defined earlier in this section. The Pareto optimal set of solutions to a MOOP may also contain some solutions that are Pareto optimal solutions and some solutions that are non-Pareto optimal solutions, but always encompass a nondominated set of solutions. It is, therefore, also possible that the set of nondominated solutions does not represent the true Pareto optimal set of solutions. For this reason, the obtained nondominated set of solutions are typically referred to as *approximately Pareto optimal solutions* in decision space which forms an *approximately Pareto front* in objective space.

3.5 Methods to compute nondominated sets of solutions

Given that the computation of high-quality nondominated set of solutions from a large set of candidate solutions may be computationally expensive, the quest for finding computational efficient methods when computing the nondominated set of solutions lying as close as possible to the true Pareto optimal set of solutions is of significant importance. There are a number of methods available in the literature to achieve this, ranging in different levels of complexity. In this section, three well-known methods are described for identifying the nondominated set of solutions for a given finite set of candidate solutions to a MOOP of the form (3.1)–(3.4). These methods include the naive and slow method, the continuous update approach and the method of Kung *et al.* [75].

3.5.1 The naive and slow method

The first method is the naive and slow method, and is considered as the most computational expensive method when computing the nondominated set of solutions. This is due to the large number of comparisons between solutions when considering a large set of candidate solutions. For example, let $\mathcal{G}$ denote the set of all candidate solutions to a MOOP, let $\mathcal{G}'$ denote the set of nondominated solutions which is also an empty set, and let i and j denote two respective solutions in set $\mathcal{G}$. As input to the naive and slow method, a solution i is supplied which must be feasible, and is then compared to the remaining solutions in the set of candidate solutions. Moreover, the comparisons between solutions are done according to the principle of solution dominance described earlier.

The method starts by comparing a solution i to a solution j, assuming that $i \neq j$, and if it is found that solution i dominates solution j, solution i is added to $\mathcal{G}'$. If, however, it is found that solution j dominates solution i, solution i is flagged which indicates that solution i may not be added to $\mathcal{G}'$. The method continues in a similar fashion until all the solutions are compared to one another. The working of the naive and slow method is illustrated in Example 3.1 and given in pseudocode form as in Algorithm 3.1.

Example 3.1 (**Naive and slow method**). *Consider the set $\mathcal{G} = \{1, 2, 3, 4, 5\}$ of candidate solutions in Figure 3.5. The algorithm starts by setting $i \leftarrow 1$ and $\mathcal{G} \leftarrow \emptyset$ (an empty set). Solution 1 is selected and compared to all the solutions in $\mathcal{G}$ which are not equal to solution 1. Starting with solution 2 and continuing with solutions 3, 4 and 5 respectively, it is clear that solution 2*

does not dominate solution 1, and similarly solutions 3, 4 and 5 also do not dominate solution 1. Solution 1 is thus not dominated by any solution in $\mathcal{G}$ and is added to the nondominated set to obtain $\mathcal{G}' = \{1\}$.

Next, i is incremented to 2 and solution 2 is now compared to all the other solutions in $\mathcal{G}$ not equal to solution 2. Selecting solutions 1, 3, 4, and 5, it is found that none of these solutions dominate solution 2 and results in solution 2 to be added to the nondominated set of solutions to obtain $\mathcal{G}' = \{1, 2\}$. Next, i is incremented to 3 and solution 3 is compared to all the other solutions in $\mathcal{G}$. It is also found that solution 3 is not dominated by any solution in $\mathcal{G}$, and solution 3 is therefore added to the nondominated set to obtain $\mathcal{G}' = \{1, 2, 3\}$.

After incrementing i to 4, solution 4 is selected and compared to the remaining solutions in $\mathcal{G}$. It is found that solution 4 is dominated by two other solutions, namely solutions 2 and 3. This results in solution 4 to be flagged and not adding it to the current nondominated set $\mathcal{G}'$. Finally, i is incremented to 5 and solution 5 is compared to all the other solutions in $\mathcal{G}$ not equal to solution 5. It is also found that solution 5 is dominated by solutions 2 and 3. Solution 5 is therefore flagged and also not added to the nondominated set of solutions. The algorithm terminates and a nondominated set $\mathcal{G}' = \{1, 2, 3\}$ is returned as output by the algorithm. □

Algorithm 3.1: **The naive and slow method [26].**

Input : A set of $\mathcal{G}$ candidate solutions to a MOOP (3.1)–(3.4).
Output: A set of nondominating solutions $\mathcal{G}' \subseteq \mathcal{G}$.

```
1  i ← 0;
2  G' ← ∅;
3  while (i ≤ |G|) do
4      j ← 0;
5      while j ≤ |G| do
6          if j ≠ i then
7              if x_j ⪯ x_i then
8                  Continue;
9              else
10                 j ← j + 1;
11     if j = |G| then
12         G' ← G' ∪ {i};
13     i ← i + 1;
14 return [G'];
```

In order to test for dominance using the naive and slow method, $\mathcal{O}(\mathcal{G})$ comparisons are required, while each comparison for dominance requires $\mathcal{P}$ objective function comparisons. This concludes that $\mathcal{O}(\mathcal{GP})$ comparisons of dominance testing are required and at most $\mathcal{O}\left(\mathcal{PG}^2\right)$ computations are performed by the algorithm.

3.5.2 The continuous update approach

The second method is the continuous updating approach which is based on the naive and slow method. This method, however, is able to find the nondominating set of solutions in a consider-

ably less computational time compared to the naive and slow method. The algorithm works in such a way that a subset of solutions are maintained throughout the execution of the algorithm which is then compared to the remaining candidate solutions using the principle of solution dominance described in §3.4. As the algorithm progresses, the subset becomes smaller until the nondominated set of solutions is uncovered. Let $\mathcal{G}$ denote the set of all the candidate solutions to a MOOP and let $\mathcal{G}'$ denote the nondominated set of solutions which is also an empty set. The algorithm is initiated by randomly selecting a solution from $\mathcal{G}$, removing it from $\mathcal{G}$ and inserting it into the empty set $\mathcal{G}'$. Each solution in $\mathcal{G}$ is then compared to each solution in $\mathcal{G}'$, one by one. In this case, let i denote a solution in $\mathcal{G}$ and let j denote a solution in $\mathcal{G}'$. If it is found that solution i dominates solution j, solution j is removed from $\mathcal{G}'$. If it is found that solution i is dominated by any solution in $\mathcal{G}'$, solution i is temporarily ignored. Moreover, if it is found that solution i is not dominated by any solution in $\mathcal{G}'$, solution i is added to $\mathcal{G}'$. The algorithm is finally terminated if all the solutions in $\mathcal{G}$ have been compared to all the solutions in $\mathcal{G}'$. This procedure ensures that the set of nondominating solutions is captured in $\mathcal{G}'$ by iteratively updating the set of solutions in $\mathcal{G}$ and $\mathcal{G}'$. The working of the continuous update approach is described in Example 3.2, and given in pseudocode form as Algorithm 3.2.

Algorithm 3.2: **The continuous update approach [26].**

Input : A set of $\mathcal{G}$ candidate solutions to a MOOP (3.1)–(3.4).
Output: A set of nondominating solutions $\mathcal{G}' \subseteq \mathcal{G}$.

```
1  i ← 0;
2  G' ← ∅;
3  while (i ≤ |G|) do
4      j ← 0;
5      while j ≤ |G| do
6          if x_i ⪯ x_j then
7              G' ← G'\ {i};
8          else if x_j ⪯ x_i then
9              Continue;
10         j ← j + 1;
11     if j = |G| then
12         G' ← G' ∪ {i};
13     i ← i + 1;
14 return [G'];
```

Example 3.2 (**The continuous update approach**). *For this example, consider again the set of* $\mathcal{G} = \{1, 2, 3, 4, 5\}$ *candidate solutions in Figure 3.5. Suppose that solution 1 is randomly selected, removed from* $\mathcal{G}$*, inserted into* $\mathcal{G}'$*, and* i *is incremented to* 2*. Solution 2 is now compared to solution 1 in* $\mathcal{G}'$*, and it is found that solution 2 does not dominate solution 1, or vice versa. This results in solution 2 to be added to* $\mathcal{G}'$*. This yields that* $\mathcal{G}' = \{1, 2\}$*, solution 2 removed from* $\mathcal{G}$*, and* i *is incremented to* 3*. Solution 3 is now compared to solution 1 and 2 in* $\mathcal{G}'$*. It is found that solution 3 does not dominate solution 1 and 2, or vice versa. Solution 3 is then added to* $\mathcal{G}'$*, removed from the set* $\mathcal{G}$*, and* i *is incremented to* 4*. Furthermore, solution 4 is now compared to solution 1, 2 and 3 in* $\mathcal{G}'$*. It is found that solution 4 is dominated by solution 2 and 3, which results in solution 4 to be ignored. Finally,* i *is incremented to* 5 *and solution 5 is now compared to solutions 1, 2 and 3. It is found that solution 5 is dominated by solutions 2 and 3, and therefore is ignored for the remainder of the iteration. This concludes that the nondominated set yields* $\mathcal{G}' = \{1, 2, 3\}$ *as output to candidate set* $\mathcal{G}$*.* □

The computational complexity of this method is considered the same as the naive and slow approach, which is $\mathcal{O}\left(\mathcal{P}\mathcal{G}^2\right)$. The number of comparisons in this method, however, is considered much less than the number of comparisons made in the naive and slow method, and is estimated to be approximately half of the number of the comparisons performed by the naive and slow method [26]. Considering the complexity of the continuous update approach, a solution in $\mathcal{G}$ is compared with one solution in $\mathcal{G}'$, followed by the next solution which is compared with at most two solutions in $\mathcal{G}'$, and so on. This yields a total of $1+2+...+(\mathcal{G}-1)$ or $\mathcal{G}\frac{(\mathcal{G}-1)}{2}$ comparisons at most.

3.5.3 Kung *et al*'s efficient method

The third and final method is the method of Kung *et al.* [75] and is considered the fastest and most efficient method when determining the nondominated set of solutions among the three methods discussed in this section. Again, let $\mathcal{G}$ denote the set of candidate solutions and let $\mathcal{G}'$ denote an empty set which is iteratively filled with nondominated solutions. The method takes as input a candidate set of solutions and generates a nondominating set of solutions from $\mathcal{G}$. The method is initiated by sorting each solution in $\mathcal{G}$ in a non-improving order according to the first objective function, and then dividing the sorted set into two halves, yielding a top subset of solutions, denoted by $\mathcal{B}$, and a bottom subset of solutions, denoted by $\mathcal{T}$. Included in subset $\mathcal{B}$ are the solutions that excels in the first objective function. Each solution i in subset $\mathcal{T}$ is then compared to each solution j in subset $\mathcal{B}$ using the notion of dominance described earlier. If a solution i in subset $\mathcal{T}$ is not dominated by any other solution in subset $\mathcal{B}$, solution i is then included into a merged set, denoted by $\mathcal{Z}$. This process is repeated with respect to $\mathcal{Z}$ until no solution in subset $\mathcal{T}$ can be found that is dominated by any other solution in $\mathcal{B}$. The subset $\mathcal{B}$, therefore, contains the set of nondominated solutions after the recursive process is terminated. The working of the method of Kung *et al.* [75] is described in Example 3.3 and given in pseudocode form as Algorithm 3.3.

Example 3.3 (**The method of Kung *et al.* [75]**). *For this example, consider again the set* $\mathcal{G}=\{1,2,3,4,5\}$ *of solutions in Figure 3.5. The candidate solutions are sorted in non-improving (decreasing) order of magnitude according to objective function* f_1 *and yields the ordered set* $\mathcal{G}=\{3,2,5,4,1\}$. *Note that the objective function* f_1 *is to be minimised. Next, the set* $\mathcal{G}$ *is partitioned into sets* $\mathcal{B}=front\left(\{3,2\}\right)$ *and* $\mathcal{T}=front\left(\{5,4,1\}\right)$ *as the size of* $\mathcal{G}$ *is five (and not one as required for termination of the algorithm). The partitioning of* $\mathcal{G}$ *to form fronts* $\mathcal{B}$ *and* $\mathcal{T}$ *is illustrated graphically in the top-most branch in Figure 3.8.*

Next, set $\mathcal{B}=front\left(\{3,2\}\right)$ *is partitioned further into sets to form* $\mathcal{B}=Front\left(\{3\}\right)$ *and* $\mathcal{T}=Front\left(\{2\}\right)$. *Since the size of both these sets is equal to 1, they are returned as output to the previous front* $\mathcal{B}=front\left(\{3,2\}\right)$.

In a similar fashion, set $\mathcal{T}=Front\left(\{5,4,1\}\right)$ *is partitioned further into sets to form* $\mathcal{B}=Front\left(\{5,4\}\right)$ *and* $\mathcal{T}=Front\left(\{1\}\right)$. *The size of set* $\mathcal{T}$ *is two which results in the further partitioning of set* $\mathcal{T}$ *into sets* $\mathcal{B}=Front\left(\{5\}\right)$ *and* $\mathcal{T}=Front\left(\{4\}\right)$. *Since the size of all the sets is 1, solutions 5, 4 and 1 are returned as output to set* $\mathcal{T}=Front\left(\{5,4,1\}\right)$.

Finally, sets $\mathcal{B}=Front\left(\{3,2\}\right)$ *and* $\mathcal{T}=Front\left(\{5,4,1\}\right)$ *are compared to one another for dominance with respect to objective function* f_2. *Following the same computational notion as with objective function* f_1, *it is found that solutions 2 and 3 dominate solutions 4 and 5 in set* $\mathcal{T}$, *and solution 1 is not dominated by any solution in set* $\mathcal{B}$. *This yields an output to the set* $Front\{3,2,5,4,1\}$ *a nondominated set of solutions* $\mathcal{Z}=front\left(\{3,2,5\}\right)$. □

Algorithm 3.3: **The method of Kung *et al.* [75].**

Input : A set of $\mathcal{G}$ candidate solutions to a MOOP (3.1)–(3.4).
Output: A set of nondominating solutions $\mathcal{G}' \subseteq \mathcal{G}$.

```
1  Sort (G);
2  if |G| = 1 then
3      return [G];
4  else
5      B ← Front (G_1, G_2, ..., G_⌊|G|/2⌋);
6      T ← Front (G_⌊|G|/2+1⌋, G_⌊|G|/2+2⌋, ..., G_|G|);
7      i ← 1;
8      Z ← ∅;
9      while i ≤ |T| do
10         j ← 1;
11         while j ≤ |B| do
12             if x_j ⋠ x_i then
13                 j ← j + 1;
14             else
15                 Continue;
16         if j = |B| then
17             Z ← Z ∪ {i};
18         i ← i + 1;
```

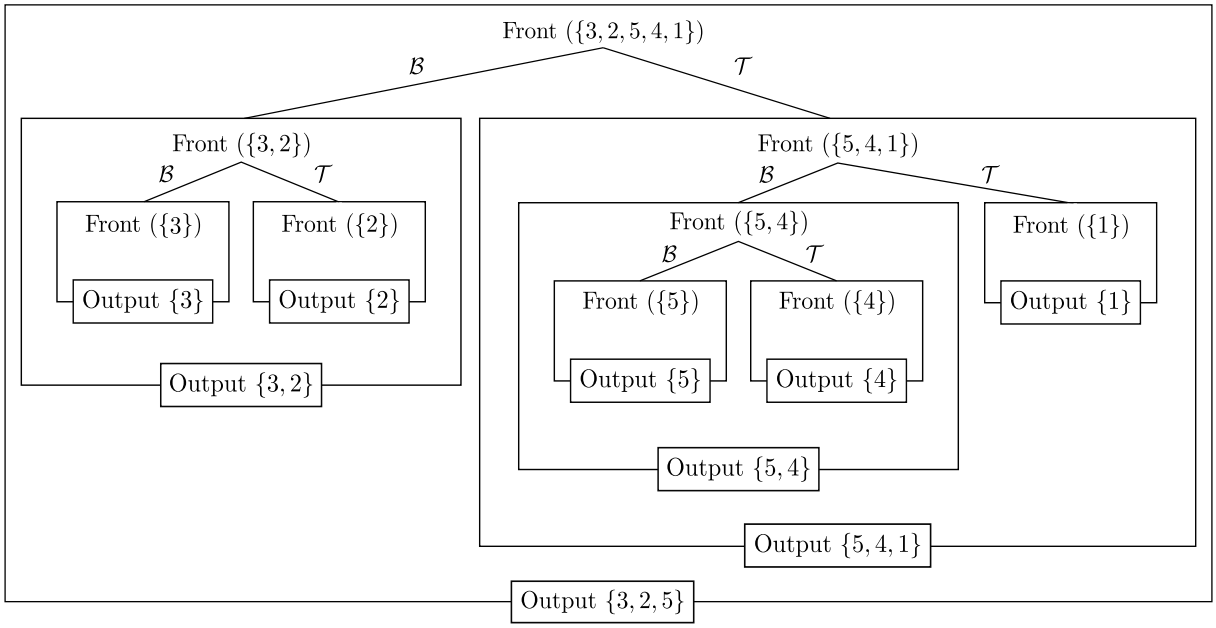


FIGURE 3.8: *A graphical illustration of the method of Kung et al. [75] for finding a nondominated set of solutions for set* $\mathcal{G} = \{1, 2, 3, 4, 5\}$ *of the five candidate solutions in Figure 3.5.*

According to Kung *et al.* [75], the computational complexity of this method is $\mathcal{O}(\mathcal{G} \log \mathcal{G})$ for $\mathcal{P} = 2$ or 3 objective functions, and $\mathcal{O}\left(\mathcal{G}(\log \mathcal{G})^{\mathcal{P}-2}\right)$ for $\mathcal{P} \geq 4$ objective functions.

3.6 A fast nondominated sorting algorithm

In general, the aim of multi-objective solution methods is to find high-quality sets of nondominated solutions within a feasible region of a problem instance. These methods employ techniques that partition a set of candidate solutions into a set of nondominated solutions and a set of dominated solutions. The nondominated set of solutions is typically returned as output to a MOOP. However, a number of algorithms demand that a set of candidate solutions be partitioned into different sets with respect to their relative degrees of dominance. In order to achieve this, a nondominated sorting algorithm may be employed, such as the *fast nondominated sorting algorithm* (FNSA) developed by Deb *et al.* [28].

In order to obtain a nondominated set of solutions among a finite set of $\mathcal{G}$ candidate solutions to a MOOP of (3.1)–(3.4), the FNSA incorporates one of the methods described in the previous section. The FNSA works in such a way that the nondominated set of solutions is called the front 1 nondominated solutions and is denoted by F_1. This set is then temporarily removed from $\mathcal{G}$ which results in a smaller (new) set of solutions remaining in $\mathcal{G}$. The nondominated set of solutions are then calculated from the new current set, and is called front 2, denoted by F_2. The solutions in F_2 are then removed from $\mathcal{G}$, which again results in a smaller (new) set of solutions remaining in $\mathcal{G}$. Again the nondominated set of solutions is calculated from the new set, which then forms front 3 and is denoted by F_3. This procedure is repeated in a similar fashion until no solutions are left in the set $\mathcal{G}$ (in other words, each solution in $\mathcal{G}$ is now partitioned in a set of nondominated solutions).

For each solution i, the number of solutions that dominate solution i is counted, known as a dominance count and denoted by dt_i. Furthermore, a set of solutions that is dominated by solution i is also computed and is denoted by S_i. The dominance count of each solution i is subjected to the nondominated front into which it is partitioned. For example, if solution i is partitioned into front F_k, then $dt_{k-1} = k-1$. The entire process requires $\mathcal{O}\left(\mathcal{P}\mathcal{G}^2\right)$ comparisons, where $\mathcal{P}$ is denoted as the number of objective functions to the MOOP, and $\mathcal{G}$ is denoted as the number of decision variables in decision space. The working of the FNSA is described by means of an example in Example 3.4 given in pseudocode form as Algorithm 3.4.

Example 3.4 (**A fast nondominated sorting algorithm**). *For this example, again consider the set of $\mathcal{G} = \{1, 2, 3, 4, 5\}$ candidate solutions in Figure 3.5. Suppose that one of the methods in §3.5 was used to obtain a nondominated set of solutions. From Examples 3.1, 3.2, 3.3, it is clear that solutions 1, 2 and 3 are nondominated solutions which result in these solutions to be partitioned into front 1. These three solutions are temporarily removed from $\mathcal{G}$ and solutions 4 and 5 remain in $\mathcal{G}$. When comparing the remaining solutions, it is found that neither of the two solutions dominated each other, and both solutions 4 and 5 are partitioned into front 2. These two solutions are also temporarily removed from $\mathcal{G}$, and results in $\mathcal{G} \to \emptyset$ followed by the termination of the algorithm. In conclusion, two nondominated fronts are obtained from the FNSA, namely F_1 and F_2, and are illustrated graphically in Figure 3.9.* □

3.7 The weighting sum of objectives

The *weighting-sum of objectives method* is regarded as one of the classical multi-objective optimisation techniques, and is considered by many authors as the simplest optimisation approach to solving MOOPs [26]. The method scalarises a set of $\mathcal{P}$ objective functions into a single objective function by incorporating user-specified weights for each objective function in (3.1). An

Algorithm 3.4: **The FNSA proposed by Deb *et al.* [28].**

Input : A set of $\mathcal{G}$ candidate solutions to a MOOP (3.1)–(3.4).
Output: A set of nondominating fronts $F_1, F_2, ..., F_k$.

```
F_1 ← ∅;
for i ∈ G do
    S_i ← ∅;
    dt_i = 0;
    for j ∈ P do
        if i ≺ j then
            S_i ← S_i ∪ {j};
        else if j ≺ i then
            dt_i ← dt_i + 1;
    if dt_i = 0 then
        i_rank ← 1;
        F_1 ← F_1 ∪ {i};
k ← 1;
while F_k ≠ ∅ do
    A ← ∅;
    for i ∈ F_k do
        for j ∈ S_i do
            dt_i ← dt_i − 1;
            if dt_i = 0 then
                j_rank ← k + 1;
                A ← A ∪ {j};
    k ← k + 1;
    F_k ← A;
```

objective that is preferred to another objective is assigned a higher weight compared to the other objective. Let w_p denote the weight assigned for objective $p \in \{1, ..., \mathcal{P}\}$, where $\sum_{p=1}^{\mathcal{P}} w_p = 1$. The MOOP of form (3.1)–(3.4) may then be scalarised to

$$\text{maximise/minimise } F(\boldsymbol{x}) = \sum_{p=1}^{\mathcal{P}} w_p f_p(\boldsymbol{x}), \tag{3.6}$$

$$\text{subject to} \quad g_a(\boldsymbol{x}) = 0, \qquad a = 1, ..., A, \tag{3.7}$$

$$h_b(\boldsymbol{x}) \geq 0, \qquad b = 1, ..., B, \tag{3.8}$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \qquad i = 1, ..., k. \tag{3.9}$$

The problem (3.6)–(3.9) is now a single-objective optimisation problem. In order to ensure that all the objectives are of the same order of magnitude, they are typically normalised before included in the objective function (3.6).

Considering the weights assigned to the objective functions, Miettinen [85] showed that if the weights $\boldsymbol{w*} = \{w_1, w_2, ..., w_p\}$ assigned to each objective function are all positive, then any optimal solution to (3.6)–(3.9) is considered a Pareto optimal solution. Furthermore, Miettinen [85]

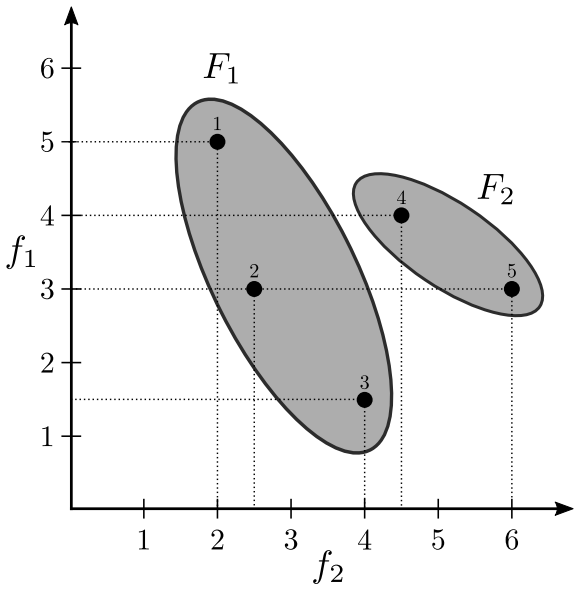


FIGURE 3.9: *The partitioning of five candidate solutions to a MOOP into two nondominated fronts using the FNSA. Each nondominated front is illustrated with a grey region surrounding the solutions included in that front.*

also showed that the converse of this result is also true if the problem in (3.6)–(3.4) is considered a convex problem. More specifically, he showed that a positive weight vector $\boldsymbol{w}* = [w_1^*, w_2^*, ..., w_p^*]$ exists if $\boldsymbol{x}^*$ is a Pareto optimal solution to a convex MOOP of the form (3.1)–(3.4) such that $\boldsymbol{x}^*$ is an optimal solution to the SOOP in (3.6)–(3.9). It, however, does not suggest that any Pareto optimal solution to the MOOP (3.1)–(3.4) can be obtained by solving an approximately weighted version of the SOOP (3.6)–(3.9). Furthermore, if the SOOP (3.6)–(3.9) is a non-convex problem, Pareto optimal solutions to the MOOP (3.1)–(3.4) exist which does not coincide with the optimal solutions to the SOOP (3.6)–(3.9) for any positive choice of the weight vector $\boldsymbol{w}$.

In order to illustrate the working of the weighted single-objective version in the SOOP (3.6)–(3.9) so as to obtain a nondominated set of solutions, consider Figure 3.10 in which the feasible region in the objective space for a bi-objective optimisation problem where both objective functions f_1 and f_2 is to be minimised. The grey region denotes the feasible region of the problem and the dark curve denotes the corresponding Pareto front. Suppose that weights w_1 and w_2 are assigned to objective functions f_1 and f_2 in the SOOP (3.6)–(3.9), and that $w_1 + w_2 = 1$. Then the objective function (3.6) is considered a convex combination of the functions f_1 and f_2 (since $\mathcal{P} = 2$), and that a straight line involving a slope $\frac{-w_1}{w_2}$ may, therefore, represent the objective function $F(\boldsymbol{x})$ to the SOOP (3.6) in the objective space. Examples of such lines are graphically illustrated in Figure 3.10 and are denoted as a, b, c and d, respectively. The location of the contour line is subjected to the objective value of the objective function $F(\boldsymbol{x})$. Since the objective is to minimise $F(\boldsymbol{x})$, the contour line may be shifted along the slope of $\frac{-w_1}{w_2}$ within the feasible region until the smallest value of $F(\boldsymbol{x})$ corresponds with the contour line. From Figure 3.10, point A on contour line d represents the Pareto optimal solution to the MOOP (3.1)–(3.4) associated with weight vector $w = [w_1, w_2]$. Assigning a different set of weights will result in a slope change of the contour, and consequently lead to another Pareto optimal solution to be uncovered. If (3.1)–(3.4) is convex, multiple Pareto optimal solutions to the MOOP (3.1)–(3.4) may be found by solving the SOOP (3.6)–(3.9) for multiple sets of positive weight vectors one at a time.

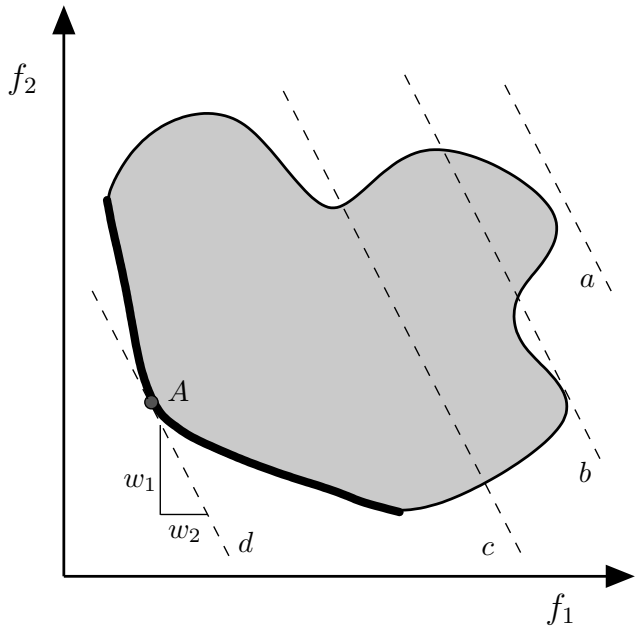


FIGURE 3.10: *Uncovering Pareto optimal solutions for a bi-objective optimisation problem where both the objective functions f_1 and f_2 are minimised using the weighted-sum approach.*

If the problem is not convex, the same procedure described above may be applied to find *certain* Pareto optimal solutions to to the MOOP (3.1)–(3.4). It is, however, possible that some of the Pareto optimal solutions are not discoverable when the MOOP (3.1)–(3.4) is not convex. For example, consider Figure 3.11 which illustrates a feasible region that is not convex and both the objective functions f_1 and f_2 are to be minimised. Suppose that weight vector w is chosen which results in contour lines a and b, and that Pareto optimal solutions A, B and C are now discoverable. Considering contour line b, it is clear that no contour line will produce a tangential point in the feasible region of points B and C. The reason for this is that before a contour line becomes tangent to any point on the line segment BC, it will also become tangent to another (improved) point in the feasible region. Therefore, no Pareto optimal solution along contour line d is computable using the weighted-sum method. It should, therefore, be noted that the weighted sum method should only be considered solving the case of convex problems and not for the case of non-convex problems [122].

3.8 Chapter summary

This chapter was devoted to an introduction to the field of multi-objective optimisation, and the aspects associated with this type of optimisation from the operations research literature. The chapter opened in §3.1 with a discussion on the computational complexity when solving optimisation problems using algorithms. This was followed by a discussion on the general formulation of MOOPs in §3.2, and includes the basic denotations of such problems. Moreover, the aspects of convex and non-convex MOOPs were discussed in §3.3, and the conclusion was made that non-convex MOOPs are more difficult to solve than convex MOOPs.

In §3.4, the notion of Pareto dominance was introduced. It was concluded that no single optimal solution exist to a MOOP, but rather that there are multiple solutions to MOOPs that are equally good. Next, in §3.5, the properties and characteristics of domination were introduced followed by the concept of Pareto optimality and strong dominance among a set of candidate solutions to

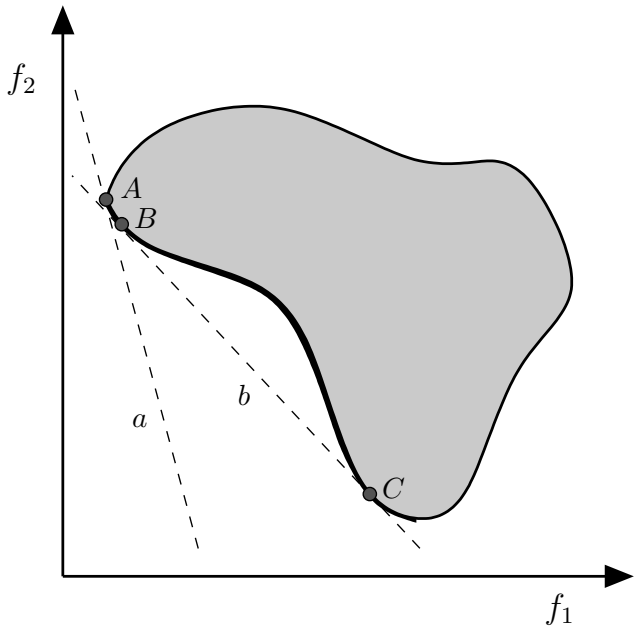


FIGURE 3.11: *Failure to uncover all the Pareto optimal solutions for a non-convex bi-objective optimisation problem where both objective functions f_1 and f_2 are to be minimised using the weighted-sum approach.*

MOOPs. Furthermore, a number of methods exist which are able to partition a set of candidate solutions to a MOOP into a nondominated set of solutions. These methods include the naive and slow approach, the continuous update approach and the method of Kung *et al.* [75].

The very well-known FNSA developed by Deb *et al.* [28] which is able to sort candidate solutions to a MOOP into different classes according to their degrees of dominance was briefly described in §3.6. In §3.7 the weighted-sum of objectives approach was described and it was found that the characteristic of the MOOP to be solved should be thoroughly analysed since the capabilities of the weighted-sum of objectives approach depend on the convexity of MOOPs. The chapter closed with a brief summary of the chapter contents in §3.8.

CHAPTER 4

Solution approaches

Contents

Optimisation problems may be solved by using one of three types of solution approaches, namely an exact solution approach, a heuristic solution approach and a metaheuristic solution approach. This chapter is dedicated to describe a number of methods which reside within each of these types of approaches.

In §4.1, the notion of exact solution approach is introduced of which two well-known exact solution methods are discussed namely the implicit *branch-and-bound* method and the explicit method of *total enumeration*. Next, in §4.2, the notion of heuristic solution approaches is introduced. Some of the disadvantages of adopting a heuristic solution approach when solving an optimisation problem is also mentioned briefly. This is then followed by a brief description on the notions of a metaheuristic solution approaches in §4.3. Four well-known metaheuristic solution methods, two of which may be employed to solve SOOPs and two of which may be employed to solve MOOPs, are discussed. The chapter closes in §4.4 with a brief summary of chapter contents.

4.1 Exact solution approaches

One way to solve combinatorial optimisation problems[1] is to follow an exact solution approach. An exact solution approach entails a method which aims to find an optimal solution within a problem specific search space by employing an exhaustive search procedure. This may be achieved either implicitly or explicitly. One advantage of adopting an exact solution approach

[1]This type of optimisation is concerned with finding an optimal or near optimal solution among a finite set of possible solutions consisting of mathematical structures, such as graphs, matroids or independence systems [46].

is that an exact (optimal) solution is returned as output to the problem, however, typically comes at a high computational expense. Exact solution approaches are, therefore, often used in environments where the total time of execution is not restricted. In the remainder of this section, two exact solution approaches namely the *branch-and-bound method* (implicit) and the *method of total enumeration* (explicit) are discussed in some detail.

The *branch-and-bound method* is a very popular method used in the literature to solve combinatorial optimisation problems in an exact manner. It was developed by Doig and Land [77] in 1970. The method is able to provide good results within acceptable computational time frames when solving optimisation problems since unsuccessful subsets of candidate solutions are disregarded simultaneously by enumerating a set of feasible candidate solutions systematically and implicitly. Suppose that an objective function $G(\boldsymbol{y})$ is to be maximised where $\boldsymbol{y}$ denotes a vector of discrete decision variables, and let $\mathcal{Z}$ denote the decision variable search space that illustrates the boundaries within which vector $\boldsymbol{y}$ may be located [113]. As the name of the method suggests, two user-specific procedures are employed namely the *branch* procedure and the *bound* procedure. The branch procedure entails returning two or more (smaller) sets $\mathcal{Z}'_1, \mathcal{Z}'_2, ...$ of a given subset $\mathcal{Z}' \subseteq \mathcal{Z}$ of the candidate solutions — the union of these sets is $\mathcal{Z}'$. As the algorithm progresses during the execution thereof, a tree-like structure is formed known as a *search tree* where the nodes in the tree represent the number of subsets in the union $\mathcal{Z}'$. For a maximisation problem, a maximum objective value for $G(\boldsymbol{y})$ is computed over the subset $\mathcal{Z}'$ which represents the maximum value for the set of solutions $\{G(\boldsymbol{y}_1), G(\boldsymbol{y}_2), ...\}$, where $G(\boldsymbol{y}_i)$ represents the minimum objective value at $\boldsymbol{y}_i$ within $\mathcal{Z}'_i$. It is important to choose a branching procedure which is able to produce non-overlapping subsets.

The bound procedure, on the other hand, is employed by the algorithm to estimate upper and lower bounds on the maximum value of $G(\boldsymbol{y})$ within a given subset $\mathcal{Z}' \subseteq \mathcal{Z}$ [113]. In the literature, there exists no universal bounding procedure although multiple methods exist which may be employed to estimate the upper and lower bounds in a situation similar to the aforementioned — that is, a specialised bounding procedure is employed for the specific problem at hand.

Suppose that the upper bound of objective function $G(\boldsymbol{y})$ for some node $\mathcal{Z}'_i$ in the search tree exists and a lower bound for another node $\mathcal{Z}'_j$ in the search tree exists, given that $i \neq j$. A global variable, denoted by l_z, is then employed which aims to keep track of the largest lower bound uncovered among all the subregions of the search space throughout the execution of the branch-and-bound method. If it found that the upper bound for node $\mathcal{Z}'_i$ is *smaller* than the lower bound for node $\mathcal{Z}'_j$, given that the optimisation problem is to be maximised, then node $\mathcal{Z}'_i$ may be entirely disregarded from the search — a node which achieves a lower upper bound value than l_z may therefore be safely disregarded from the search tree. This procedure is known as *pruning* [103, 113].

A stopping criteria is also employed in the algorithm which indicates when the algorithm should terminate. A typical stopping criteria that may be implemented is to terminate the search once the current candidate set $\mathcal{Z}$ is reduced to a single element, or when the objective function upper bound of the set $\mathcal{Z}$ corresponds to any available objective function lower bound. In such a case, the value of the objective function $G(\boldsymbol{y})$ has achieved a maximum value within $\mathcal{Z}$ at any element of $\mathcal{Z}$.

Another method that may be used to solve combinatorial optimisation problems is the method of *total enumeration*. This method merely enumerates through all the possible candidate solutions to the problem in an iterative fashion while keeping track of the solutions that achieved the best objective function value thus far during the search [103]. One disadvantage of employing this method is that inordinate computational time may be required to solve such a problem in

an exact manner. Within a military based environment, the weapon assignment problem[2] for example is a case where an optimal solution is required within a short time frame. A *heuristic solution approach* may therefore be employed to solve large combinatorial optimisation problems. These solution approaches are able to provide good quality solutions (not necessarily exact), but in much shorter time frames than exact solution approaches. The notion of heuristics is discussed in some detail in the next section.

4.2 Heuristic solution approaches

The word *heuristic* stems from the Greek word *heuriskein* which means to *find* or *discover*. In this sense, heuristics are very popular methods for solving optimisation problems and are in nature very different to exact solution approaches. This is due to the fact that near-optimal solutions are computed for optimisation problems rather than optimal solutions given that the optimisation problem is too complex to solve exactly [61]. The goal in employing heuristic solution approaches is to search for high-quality solutions with a substantial reduced computational time when compared to the execution time of exact solution approaches. There exists three well-known classes of heuristics in the operations research literature. The first class includes *local search algorithms* which make use of a hill climbing technique to discover good quality solutions whereas the second class includes *constructive algorithms* which are mainly algorithms that implement a greedy incremental solution construction approach. The third and final class includes *iterative algorithms* which iteratively evolve over time in pursuit of high-quality solutions.

Local search algorithms generate a random complete initial feasible candidate solution, and then iteratively change a single element in the solution vector in order to uncover a sequence of improved successive solutions. Here, a solution is only accepted as a new solution if a change in the solution vector results in an improvement in the objective function. Continuing in a similar fashion, a number of changes are then performed on the solution vector during every iteration to obtain a number of improvements. If, however, a change in the solution vector results in a worsening objective function, an alternative change is performed on the solution vector until an improvement is found. The algorithm is therefore iterated until no further improvements can be found [61].

A constructive heuristic approach, on the other hand, follows a greedy approach towards selecting solutions, and therefore results in the selection of solutions which yield the best improvement in the current iteration without considering future consequences. This heuristic approach is different from the local search heuristic approach in the sense that the algorithm starts with an empty solution, and then repeatedly extends on this solution until a complete solution is computed. Constructive heuristic algorithms require little computational time to obtain a good candidate solution once they terminate, however, are often regarded as far from optimal [61].

Finally, iterative algorithms are defined as algorithms which evolve over a number of iterations in a quest to uncover an improved solution. These approaches are regarded as simple and robust procedures which usually yield solutions that are inferior in quality. Expert or intuitive knowledge disguised as rules of thumb that governs the search procedure are typically incorporated in these approaches. These methods also involve multiple stopping criteria that must be achieved before the algorithm may be terminated [61].

[2]Within the operations research literature, the underlying combinatorial optimisation problem where available weapon systems are assigned to threats is known as a weapon assignment problem [79].

Some of the disadvantages associated with heuristic approaches are as follows. They are typically problem specific and exhibit unadaptable characteristics. They also portray greedy behaviour in terms of solution selection which may result in the algorithm becoming trapped in local optima. Due to the aforementioned disadvantages, included but not limited to, multiple disadvantages have been associated with heuristic solution approaches which resulted in introducing *metaheuristics*. These metaheuristics are considered to outperform heuristic solution approaches in many problem instances, and are briefly described in the next section.

4.3 Metaheuristic solution approaches

The development of *metaheuristics* solution approaches was substantiated by a number of disadvantages associated with heuristic solution approaches described earlier. The addition of "meta" in *metaheuristics* means *beyond* or *superior level*, implying that a metaheuristic approach (in general) is able to perform better when compared to a heuristic approach (pointing to the ability to avoid getting stuck in local optima). According to Hillier and Lieberman [61], a metaheuristic solution approach is one that coordinates the interaction between local improvement procedures and higher level strategies to create an unique characteristic in the sense that a robust search is performed on the feasible region without getting trapped in local optima. Metaheuristics may typically be applied to more than one type of problem, given that they provide guidelines and general structure with respect to the formulation of a tailored solution approach to problems.

Metaheuristic approaches may be divided into two distinct classes, namely *trajectory-based methods* and *population-based methods*. In trajectory-based approaches, a single candidate solution is maintained and improved throughout the search for a global optimal solution, where in population-based approaches, a population of candidate solutions are iteratively changed at once in the quest to uncover high quality local optimal solutions. Examples of trajectory-based metaheuristics solved for single-objective optimisation problems include the method of *simulated annealing* (SA) proposed by Kirkpatrick *et al.* [73], the *tabu search* proposed by Glover [56], the *variable neighbourhood search* proposed by Hansen and Mladenovic [87], and the method of *harmony search* proposed by Geem *et al.* [53]. Examples of population-based metaheuristics include the *genetic algorithm* (GA) proposed by Holland [63], the *ant colony optimisation algorithm* proposed by Dorigo *et al.* [37], and the *particle swarm optimisation algorithm* proposed by Kennedy and Eberhart [98].

	Single-objective optimisation problem	**Multi-objective optimisation problem**
Trajectory-based	The method of simulated annealing	Dominance based multi-objective simulated annealing
Population-based	The genetic algorithm	Nondominated sorting genetic algorithm II

TABLE 4.1: *A classification framework for solution approaches towards solving single-objective and multi-objective optimisation problems.*

Metaheuristic solution approaches may further be extended to include solution approaches for solving MOOPs of the form (3.1)–(3.4). Extensions of the aforementioned single-objective solution approaches include the *multi-objective genetic algorithm* proposed by Parks [95], the *multi-objective simulated annealing* (MOSA) algorithm designed by Engrand [43], the *vector evaluated particle swarm optimisation* originally proposed by Parsopoulos and Vrahatis *et al.* [96], the *nondominated sorting genetic algorithm II* (NSGA II) proposed by Agarwal *et al.* [28] and the

dominance based multi-objective simulated annealing (DBMOSA) algorithm proposed by Smith *et al.* [115]. The NSGA II and the DBMOSA are extensions of the GA and the method of simulated annealing when solving multi-objective optimisation problems. The classification of metaheuristic solution approaches for solving single-objective and multi-objective optimisation problems, together with examples of the solution approach classes are illustrated in Table 4.1.

In the remainder of this section, two single-objective solution approaches *i.e.* the method of simulated annealing and the GA, and two multi-objective solution approaches *i.e.* the NSGA II and the DBMOSA are discussed in some detail.

4.3.1 The method of simulated annealing

The method of *simulated annealing* was proposed by Kirkpatrick *et al.* [73] in 1983 and is based on the fundamental principles in statistical mechanics where an annealing process is applied to solids in order to strengthen the properties of metallic materials. The method of simulated annealing resides within the fields of computational intelligence and metaheuristics, and is generally described as a stochastic optimisation algorithm involving a combination of local and global search techniques [14].

The annealing process mentioned above entails the heating of a metal above its recrystallization temperature, which is then followed by a slow cooling process. When the metal is heated, the atoms become excited and start to randomly vibrate through higher energy states [7]. When the metal is then slowly cooled, the vibration frequency of the atoms decreases until a low energy state is reached. The slow cooling procedure increases the chance for atoms to reside in a lower energy state rather than residing in the initial state.

Simulated annealing algorithms are also able to control the phenomenon of *cycling*[3], which is typically induced by accepting non-improving moves in local searches which is governed by probabilities that are randomly generated. This method is considered as a trajectory-based method since it iteratively performs operations on a single solution at a time. A pseudocode listing of the working of the method of simulated annealing is given for a minimisation problem as Algorithm 4.1.

The simulated annealing algorithm requires an initial feasible solution vector $\boldsymbol{x}_0$, the maximum number of iterations i_{max} to execute throughout the search, and an initial temperature T_0 as input for initialisation. The initial solution vector is taken as the incumbent solution when the algorithm is initialised. During each iteration i of the algorithm, a neighbouring solution $\boldsymbol{x}'$ is generated from the current solution $\boldsymbol{x}$ by using a *neighbourhood move operator*. The move operator typically performs a random perturbation on the current solution $\boldsymbol{x}$ according to a set of possible moves — the set of possible moves typically takes into account the specific combinatorial context of the problem.

A move that results in an improvement in the objective function value of $\boldsymbol{x}'$ (*i.e.* $f(\boldsymbol{x}') < f(\boldsymbol{x})$ for a minimisation problem) is always accepted whereas a move that does not improve the objective function value of $\boldsymbol{x}'$ is accepted with some probability. Accepting a worsening

[3]This phenomenon was introduced as part of the Simulated Annealing framework and bind the convergence properties of the algorithm to a small number of simple parameters which describe the geometry of the energy landscape [29].

neighbouring solution is governed by the well-known *Metropolis-Hastings rule* [73], which states that a worsening neighbouring move be accepted with a probability

$$\exp\left(\frac{-\triangle_{obj}}{T_i}\right), \tag{4.1}$$

where $\triangle_{obj}$ denotes the change in objective function value when performing a neighbouring move operator on the current solution $\boldsymbol{x}$, and T_i denotes the temperature for iteration i. If the neighbouring solution is accepted, it becomes the new current solution for iteration $i+1$ of the algorithm. When a neighbouring solution is rejected, however, the current solution $\boldsymbol{x}$ remains as is for the next iteration $i+1$. Moreover, if the neighbouring solution performs better than the current incumbent solution $\boldsymbol{x}_b$, $\boldsymbol{x}'$ is taken as the new incumbent solution $\boldsymbol{x}_b$. This process is iterated until i_{max} is reached.

Algorithm 4.1: The method of simulated annealing for an optimisation problem in which the objective function is to be minimised [14, 73].

Input : An initial feasible solution $\boldsymbol{x}_0$, the maximum number of iterations $i_{\max}$, and an initial temperature T_0.

Output: An approximation of the minimum objective function value and the solution vector $\boldsymbol{x}_b$ that minimises the objective function found in the search.

```
1  x_b = x_0, x = x_0;
2  for i = 1 to i_max do
3      Generate neighbouring solution x';
4      Generate random number r ∈ (0, 1);
5      if △_obj < 0 then
6          x ← x';
7          if f(x') < f(x_b) then
8              x_b ← x';
9      else if r < e^(−△_obj/T_i) then
10         x ← x';
11     If a sufficient number of iterations have passed since the last temperature change,
       reduce temperature T_i;
12 Return x_b;
```

When T_i is large, predominant neighbouring solutions are easily accepted iteratively as the new current solution which ensures that the search does not get trapped in a local optima. It also encourages the discovery of near optimal solutions by accepting worsening solutions. When T_i is small, on the other hand, a neighbouring solution that result in a small decrease in the objective function value is accepted. The algorithm, therefore, takes a large value of T_0 as input to allow for as much exploration as possible within the decision space during the early stages of the algorithm. It may, therefore, be said that the temperature T_i governs the randomness of the search.

According to Busetti [15], the value of T_0 should ensure that approximately 80% of all the worsening neighbouring solutions are accepted in the beginning of the search. This may be achieved by a random walk conducted over the decision space of the problem where the initial

feasible solution x_0 is taken as the starting point. The initial temperature may then be calculated as

$$T_0 = \frac{-\triangle^+_{obj}}{\ln 0.8}, \tag{4.2}$$

where $-\triangle^+_{obj}$ denotes the average increase in the objective function value for a fixed number of worsening neighbouring solutions accepted during the random walk.

During the execution of the algorithm, the temperature remains constant for a number of consecutive iterations, and is known as an *epoch*. The length of an epoch is associated with a maximum number of iterations rather than taken as a fixed parameter value. A Markov chain is typically used to determine the length of an epoch, denoted by L_i. According to Busetti [15], this estimate should not be a function of the iteration i but rather be customised according to the optimisation problem that is being solved. It seems intuitive to define A_{min}, which is the minimum number of move acceptances during any epoch before the temperature is lowered and the next epoch is initiated. This parameter is taken as a pre-specified parameter beforehand. As the algorithm progresses, the acceptance probability $e^{\left(-\triangle_{obj}/T_i\right)}$ lowers as the temperature T_i approaches zero, and results in the number of trails expected before accepting A_{min} moves to become larger (without bound) as the search progresses, irrespective of the value of A_{min}. An epoch is terminated once S moves have been attempted (this is when the temperature is increased) or A_{min} moves have been accepted (this is when the temperature is lowered), where $S > A_{min}$. Dreo *et al.* [38] proposed a rule of thumb for estimating the length of an epoch, and suggests that $S = 100N$ and $A_{min} = 12N$ where N denotes a measure of the number of degrees of freedom of the optimisation problem.

Cooling and reheating are employed by the algorithm and may occur many times throughout the execution of the algorithm. This is typically dependant on the number of epoch iterations that takes place. The temperature is cooled in order to reduce the acceptance probability and, hence, making it harder to accept worsening neighbouring solutions (this is done in order to promote exploitation), whereas the temperature is reheated in order to increase the acceptance probability such that worsening neighbouring solutions are accepted more frequently (this is done in order to promote exploration by allowing the algorithm to escape from local optima).

A number of cooling schedules exist in the literature that may be employed to achieve this. Examples of such cooling schedules include a geometric, an adaptive, and a linear cooling schedule [7, 92, 130, 138]. The geometric cooling schedule is regarded as the most popular cooling schedule and involves simply reducing the temperature for each iteration of the algorithm with a fixed factor α [138]. The value of α is typically chosen between 0 and 1. The temperature during a next iteration T_{i+1}

$$T_{i+1} \leftarrow \alpha \times T_i, \qquad i = 0, 1, 2, ..., i_{\max}. \tag{4.3}$$

According to Vigeh [138], α may be chosen between $0.8 \leq \alpha \leq 0.99$ to achieve high-quality results, however, according to Abdullah [1], α should be chosen $\alpha > 0.83$. A large α results in a faster cooling rate which ensures that the algorithm converges faster towards a local optima. A fast cooling rate, however, may result in a deterioration of the quality of the solutions. Furthermore, the aim of the study done by Abdullah [1] was to find quality solutions for a large set of attributes. This may, therefore, explain why Abdullah recorded α to be larger than 0.83 to achieve high-quality results.

The linear cooling schedule, on the other hand, involves a linear approach towards reducing the temperature during each iteration, and is merely reduced by a constant. The temperature during each iteration in the linear cooling schedule may be estimated as

$$T_{i+1} \leftarrow T_i - temp, \qquad i = 0, 1, 2, ..., i_{\max}, \tag{4.4}$$

where *temp* denotes a constant cooling factor. Finally, the adaptive cooling schedule, proposed by Huang *et al.* [105] in 1986, entails an exponential decrease in the temperature during each iteration. The temperature for the adaptive cooling schedule during each iteration may be estimated as

$$T_{i+1} \leftarrow T_i \times \exp\left(\frac{-\omega T_i}{\sigma(T_i)}\right), \qquad i = 0, 1, 2, ..., i_{\max}, \tag{4.5}$$

where ω denotes an empirical derived parameter chosen as $\omega \in (0, 1]$, and $\sigma(T_i)$ denotes the standard deviation of the changing values in the objective function since the start of epoch i. According to Huang *et al.* [105], ω is typically chosen as 0.7.

The notion of reheating may also be employed in a similar fashion as cooling, except that the temperature is increased by some factor. According to Abdullah [1], a reheating procedure where the current temperature is reheated to the initial temperature T_0 is more successful in exploring larger areas in the decision space. Other reheating schedules entail the geometric reheating schedule where the temperature during each epoch is increased by a fixed value β, typically chosen larger than 1. Following the same notation as the cooling schedules in (4.3)–(4.5), the temperature during epoch $i + 1$ may be estimated as

$$T_{i+1} \leftarrow \beta \times T_i, \qquad i = 0, 1, 2, ..., i_{\max}. \tag{4.6}$$

For a linear reheating schedule, the temperature during each epoch is increased by adding a reheating factor to the current temperature. The temperature for epoch $i + 1$ may then be estimated as

$$T_{i+1} \leftarrow T_i + temp_r, \qquad i = 0, 1, 2, ..., i_{\max}, \tag{4.7}$$

where $temp_r$ denotes the reheating factor. As with the linear cooling schedule (4.4), the temperature is increased with the same amount during the execution of the algorithm. The procedure of reheating and cooling is iterated until a stopping criteria is reached where the incumbent solution $\boldsymbol{x}_b$ is then returned as output to the algorithm.

In conclusion, the method of simulated annealing aims to find a high-quality locally optimal solution with the least amount of computational effort. This is achieved by supplying a tailored set of input parameters for every optimisation problem instance where the input parameters include the initial temperature, the length of an epoch, a cooling and reheating schedule, and a stopping criteria. One popular stopping criteria entails the specification of a maximum allowable number of epochs while another stopping criteria entails terminating the algorithm when the temperature is close to zero for an extended period of time — this allows the algorithm to converge towards a locally optimal solution.

4.3.2 The genetic algorithm

The GA was originally proposed by Holland [63] in 1975 and is based on Darwin's theory of evolution. The algorithm belongs to the field of evolutionary computation[4] and resides in the realm of evolutionary algorithms [14]. However, according to Schlunz [110], this algorithm only became popular in 1989 after the seminal work done by Goldberg [57].

In the GA, a population of individual candidate solutions (also known as *chromosomes*) are allowed to evolve over time by using evolutionary operators in a bid to uncover near-optimal solutions to an optimisation problem instance. During each iteration, a *fitness* value is assigned to each solution in the population which is an indication of the quality of the solution when compared to other solutions in the population, and is typically computed by using the objective function in the problem. Parent solutions in the population are then selected according to their fitness values in order to populate the next generation of candidate solutions, known as *offspring* solutions. Offspring solutions are typically produced by employing a *selection* operator to the current population for selecting parent solutions, which is then followed by *recombination* operators that is applied to the selected parent solutions in order to produce offspring solutions. The next generation of candidate solutions then constitutes the offspring solutions. This procedure is repeated over a number of solution generations until no more significant fitter solutions are found, or until a pre-specified number of generations is reached. A pseudocode listing of the working of the GA is given as Algorithm 4.2.

Algorithm 4.2: GA for an optimisation problem in which the objective function is to be minimised or maximised [14, 63].

Input : The population size n, probability for crossover p_c, probability for mutation p_m, maximum number of iterations $i_{\max}$

Output: Solution vector $\boldsymbol{x}_b$ that minimises the objective function

```
1  P^(0) ← GeneratePopulation(n, p_s);
2  EvaluateFitness(P^(0));
3  x_b ← GetBestSolution(P^(0));
4  i = 1;
5  while i ≤ i_max do
6      Parents ← SelectParents(P^(i), n);
7      Children ← ∅;
8      for Parent_1, Parent_2 ∈ Parents do
9          Child_1, Child_2 ← Crossover(Parent_1, Parent_2, p_c);
10         Children ← Mutation(Child_1, p_m);
11         Children ← Mutation(Child_2, p_m);
12     EvaluateFitness(Children);
13     x_b ← GetBestSolution(Children);
14     P^(i) ← Replace(P^(0), Children);
15     i = i + 1;
16 Return P^(i), x_b;
```

The GA requires as input the size of the population n, the maximum number of iterations i_{max} for the search to execute, the probabilities for recombination p_c and p_m, and the tour size S_t that

[4] In computer science, evolutionary computation is considered as a subclass of artificial intelligence whose principles are based on biological evolution theory. These techniques are mainly known as population-based solution methodologies applied to problems in the field of optimisation [47].

is employed in the selection procedure. The algorithm is initiated by providing an initial feasible population of candidate solutions $\boldsymbol{P}^{(0)}$ of size n which is typically generated randomly. This is then followed by the `EvaluateFitness` function which calculates the fitness of each candidate solution in the population. Next, the selection operator and the recombination operators *i.e. crossover* and *mutation* are applied to the population, called the `SelectParents`, `Crossover` and `Mutation` functions [103].

The selection operator first selects two parent solutions for crossover. A popular method to achieve this is the so-called *tournament selection* procedure where a small subset of candidate solutions is randomly chosen from the population [4]. The subset is called a *tour* where the size of the tour (*i.e.* the number of solutions partaking in the tour) is called the *tour size*. A solution is then selected from the subset and placed into a *mating pool* where the number of solutions in the mating pool is known as the *pool size*. The selection of parent solutions for inclusion in the mating pool are typically governed by their respective fitness values (*i.e.* a parent solution with a high objective function value is more desirable for inclusion in the case of a maximisation problem). A simple method called the *roulette wheel selection* procedure is employed in order to achieve this [97]. This method calculates a probability for a solution of being selected according to its fitness value as

$$p_a = \frac{f_a}{\sum_{b=1}^{M} f_b}, \tag{4.8}$$

where f_i denotes the fitness value assigned to solution i. Next, a circular wheel is then partitioned into arcs where each arc represents each solution, and the arcs spanning angles are proportional to the fitness value of the corresponding solution. A fixed point is then selected on the roulette wheel and rotated (similar to a roulette wheel in a casino). The solution in the arc of the wheel which corresponds to the fixed point is then chosen as the first parent solution for crossover and then removed from the population. This procedure is then repeated in a similar fashion to select a second parent solution. One advantage of this approach is that the fitness values of the solution is normalised throughout.

Once two parent solutions are selected, the crossover operator is employed to generate offspring solution(s). This operator is stochastic in nature and the probability of occurring is denoted by p_c. The crossover procedure is considered to bring diversity among the newly generated population, and therefore the probability p_c is typically chosen large [61, 74, 103]. A number of crossover operators exist in the literature, and include the single-point and two-point crossover methods, the uniform crossover method and the cut-and-splice crossover method [61, 132, 109]. The single-point crossover entails selecting a *single* point along the parent solution encodings by means of a uniform distribution. The encodings are then sliced at the selection point of which the *offspring* solutions are then generated by interchanging the different parts of the solution encodings. The notion of crossover using a single-point procedure is illustrated graphically in Figure 4.1.

A two-point crossover procedure, on the other hand, entails the selection of *two* points alongside the parent solution encodings, also by means of a uniform distribution. The offspring solutions is then generated by interchanging the parent solution encodings now varying in length. Moreover, a uniform crossover procedure entails the selection of elements in the encoding of the parent solutions governed by probabilities. The selection of a parent solution is not considered as a whole, but rather elements in the encoding is considered individually, forming a biased selection criteria when generating offspring solutions.

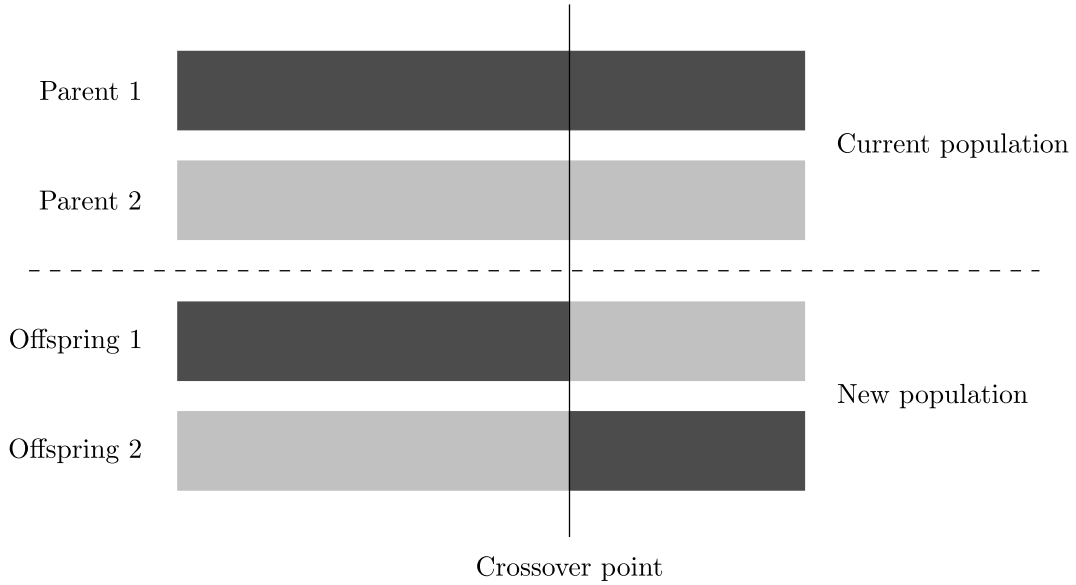


FIGURE 4.1: *An illustration of the single-point crossover procedure performed on two parent solutions to generate two offspring solutions for the next population in the GA.*

Next, mutation is applied to some of the offspring solutions generated from the crossover procedure. The goal of mutation is to ensure diversification among solutions in order to encourage the exploration of new regions in the solution space without converging towards poor local optima. Mutation typically involves altering a single or more entries in the original solution encoding — a small change is usually made. The procedure is also randomly executed with a probability p_m of occurring and is also considered a stochastic operator [61]. The probability for mutation is usually taken as small since a high probability may lead to a random search in the GA [74]. A number of mutation operators exist in the literature including a uniform mutation, Gaussian mutation and the bit flip method [132, 109].

The uniform mutation process entails the random selection of a bit in the offspring solution encoding, which is then replaced with a uniform random value governed by a pre-specified user upper and lower bound. However, this method is only applied to problems that involves integer-coded solutions or real-number solutions. The Gaussian method implements the same bit selection procedure as used in the uniform mutation method, however, a Gaussian distributed random value is added to the selected bit rather than replacing the bit. If the addition results in the bit to fall outside the user-specified upper and lower bounds, however, the bit is assigned the pre-specified upper or lower bound where applicable. Finally, the bit flip method is employed when solution encodings are binary in nature. This procedure entails inverting a random selected bit (*i.e.* flipping the bit from 0 to 1 or *vice versa*). The bit flip method is graphically illustrated in Figure 4.2 where the sixth gene was randomly selected and flipped from 0 to 1.

Offspring before mutation [1 0 1 1 0 **0** 1 0]
↓
Offspring after mutation [1 0 1 1 0 **1** 1 0]

FIGURE 4.2: *The bit fit mutation operator applied to a binary encoded offspring solution during the mutation process in the GA.*

The GA also incorporates a stopping criterion during the execution of the algorithm. Typical stopping criteria include a predefined number of iterations, achieving a satisfactory level of fitness for the population of solutions, or when the algorithm is unable to sufficiently improve

the quality of solutions in the population [61, 74, 103]. Once the algorithm is terminated, the best solution in the population is returned as output.

4.3.3 The dominance-based multi-objective method of simulated annealing

The DBMOSA is a metaheuristic solution approach for solving multi-objective optimisation problems of the form (3.1)–(3.4), and was proposed by Smith *et al.* [115] in 2008. Many of the principles found in the method of simulated annealing is employed in the DBMOSA since this metaheuristic is merely an extension on the method of simulated annealing, as mentioned in the introduction of this section. The main difference between the method of simulated annealing and the DBMOSA, however, is the notion of *archiving*.

The notion of archiving

When solving a single-objective optimisation problem by means of the SA algorithm, a single solution is maintained throughout the search and iteratively compared to the current uncovered optimal solution. For a MOOP, on the other hand, an external set of nondominated solutions, called the *archive* and denoted by $\mathcal{A}$, is maintained throughout the search procedure and compared to all the solutions within the archive before inclusion. Furthermore, the notion of solution dominance described in §3.4 is employed when comparing a neighbouring solution with the current solution in $\mathcal{A}$ before including the solution in $\mathcal{A}$. The notion of archiving is graphically illustrated in Figure 4.3 for a bi-objective optimisation problem in which both objective functions are to be maximised.

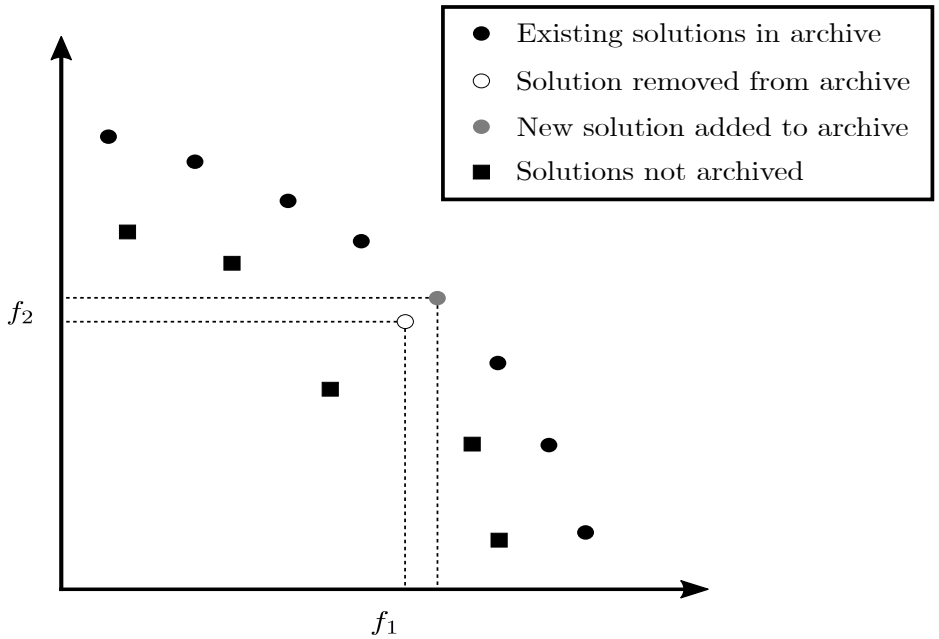


FIGURE 4.3: *A graphical illustration of the notion of archiving in an optimisation problem where two objective functions f_1 and f_2 are to be maximised.*

In Figure 4.3, the current set of nondominating solutions in $\mathcal{A}$ is represented by the black circles, the neighbouring solutions which are not included in the archive are represented by the black squares, the new neighbouring solution is represented by the grey circle, and the current solution that is removed from $\mathcal{A}$ is represented by the open circle. From the figure, it is clear that the

neighbouring solution (grey circle) dominates a solution in $\mathcal{A}$ (open circle), which results in the dominated solution to be removed from the nondominated set of solutions in $\mathcal{A}$ and the inclusion of the neighbouring solution in $\mathcal{A}$. In a similar fashion, a new neighbouring solution is iteratively generated and compared to the set of nondominated solutions in $\mathcal{A}$ in a bid to uncover a Pareto set of optimal solutions to the optimisation problem.

Let $\boldsymbol{x'}$ denote a neighbouring solution to a MOOP of form (3.1)–(3.4) and let $\boldsymbol{x}$ denote the current solution to a MOOP of form (3.1)–(3.4). Recall from §4.3.1, that the sign of the difference in energy $\triangle_{obj}$ provided information on the quality of the neighbouring solution $\boldsymbol{x'}$ with respect to the current solution $\boldsymbol{x}$ — it indicates whether the neighbouring solution $\boldsymbol{x'}$ performs better or worse than the current solution $\boldsymbol{x}$ in terms of the objective function values. For a multi-objective problem, suppose that the theoretical true Pareto front $\mathcal{P}_F$ is known. It would then be possible to define the energy of a solution $\boldsymbol{x}$ as a measure of the portion of the true Pareto front that dominates $\boldsymbol{x}$. The energy of $\boldsymbol{x}$ may then be defined as

$$E(\boldsymbol{x}) = \mu(\mathcal{P}_F(\boldsymbol{x})), \tag{4.9}$$

where μ denotes a measure defined on $\mathcal{P}_F$, and $\mathcal{P}_F(\boldsymbol{x})$ denotes the portion of the true Pareto front. The portion of the true Pareto front is then

$$\mathcal{P}_F(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathcal{P} \mid \boldsymbol{y} \prec \boldsymbol{x}\}. \tag{4.10}$$

Note that $\mathcal{P}_F$ may either be discrete or continuous. In the case where $\mathcal{P}_F$ is discrete, the cardinality of $\mathcal{P}_F(\boldsymbol{x})$ may replace $\mu(\mathcal{P}_F(\boldsymbol{x}))$ — the number of solutions which forms part of $\mathcal{P}_F$ and dominates $\boldsymbol{x}$. In the case where $\mathcal{P}_F$ is continuous, μ may be taken as a Lebesgue measure — the length, area or volume of the Pareto front which dominates $\boldsymbol{x}$ for the cases of two, three or four objective functions, respectively.

It should be noted, however, that the true Pareto front is seldom known during the optimisation process, and thus renders the use of the energy function in (4.9) when estimating the energy rather difficult. Smith *et al.* [115] defined the energy function in terms of the current estimate of the Pareto front — that is the set of nondominating solutions that has been uncovered thus far during the search procedure (*i.e.* solutions that are typically contained in $\mathcal{A}$). As a result, the difference in energy may be estimated by differentiating between the energy of the current solution in $\mathcal{A}$ and the energy of the neighbouring solution. This energy difference is then normalised by the size of the archive denoting $|\mathcal{A}|$. Smith *et al.* [115] claimed that the use of this energy measure promotes convergence towards the true Pareto front as well as the coverage of the true Pareto front.

The working of the DBMOSA is given in pseudocode form as Algorithm 4.3. The algorithm is initiated by placing an initial feasible solution $\boldsymbol{x}$ that is generated randomly into the archive $\mathcal{A}$. Next, a neighbouring solution $\boldsymbol{x'}$ is generated from the neighbourhood of $\boldsymbol{x}$, denoted by the `perturb(x)` function, where the neighbouring solution is then included in an additional archive, defined as $\tilde{\mathcal{A}} = \mathcal{A} \cup \{\boldsymbol{x}\} \cup \{\boldsymbol{x'}\}$. Moreover, define $\tilde{\mathcal{A}}_{\boldsymbol{x}} = \left\{\boldsymbol{y} \in \tilde{\mathcal{P}} \mid \boldsymbol{y} \prec \boldsymbol{x}\right\}$ where $\mu(\tilde{\mathcal{A}}_{\boldsymbol{x}}) = |\tilde{\mathcal{A}}_{\boldsymbol{x}}| + 1$, which implies that $\tilde{\mathcal{A}}_{\boldsymbol{x}}$ is a subset of all the solutions in $\mathcal{A}$ that dominates $\boldsymbol{x}$. The energy difference between $\boldsymbol{x'}$ and $\boldsymbol{x}$ may then be estimated as

$$\triangle_E(\boldsymbol{x'}, \boldsymbol{x}) = \frac{|\tilde{\mathcal{A}}_{\boldsymbol{x'}}| - |\tilde{\mathcal{A}}_{\boldsymbol{x}}|}{|\tilde{\mathcal{A}}|}. \tag{4.11}$$

Algorithm 4.3: Dominance-based multi-objective simulated annealing [115].

Input : An instance of a MOOP of the form (3.1)–(3.4), an initial feasible solution $\boldsymbol{x}$, the maximum number of epoch iterations j_{max}, the minimum number of moves accepted per epoch A_{min}, the cooling function used to determine the new temperature, the maximum number of iterations that the algorithm may execute i_{max}.

Output: A nondominated set of solutions $\mathcal{P}_S$ which is an approximation of the Pareto front for the instance (3.1)–(3.4).

```
1  Generate initial feasible solution x;
2  Initialise archive A = {x};
3  Initialise the number of iterations i ← 1;
4  Initialise the number of epochs j ← 1;
5  Initialise the number of epochs without accepting a solution γ ← 1;
6  while i ≤ i_max do
7      A ← 0;
8      while j ≤ j_max & A < A_min do
9          Generate neighbouring solution x' ← perturb(x);
10         Assess the energy difference △_E(x', x);
11         Generate random number r ∈ (0, 1);
12         if r < min(1, exp(△_E(x', x) / T_i)) then
13             x ← x';
14             if |A_x| = 0 then
15                 A ← A ∪ {x};
16                 for y ∈ A do
17                     if x ≺ y then
18                         A ← A \ {y};
19                         A ← A + 1;
20         j = j + 1;
21     T_i ← UpdateTempareture(T_i);
22     if A = 0 then
23         γ = γ + 1;
24     i ← i + 1;
25 P_S ← A;
```

Dividing the energy difference between $\boldsymbol{x}'$ and $\boldsymbol{x}$ by the magnitude of archive $\tilde{\mathcal{A}}$ ensures that the estimated energy difference $\triangle_E(\boldsymbol{x}', \boldsymbol{x})$ remains below unity. An advantage of this is that it may provide soundness against fluctuations in the number of solutions contained within $\mathcal{A}$ during the search. Implicitly, if $\tilde{\mathcal{A}}$ is a nondominated set, the energy difference between any two solutions contained in $\tilde{\mathcal{A}}$ is zero. Hence, $\triangle_E(\boldsymbol{x}', \boldsymbol{x}) < 0$ when $\boldsymbol{x}' \prec \boldsymbol{x}$ since the current as well as neighbouring solution are included in the energy function in (4.11). Another advantage of using the energy function in (4.11) is that regions of the true Pareto front which are sparsely populated are encouraged to be explored, regardless of the portion of the true Pareto front that dominates $\boldsymbol{x}$ and $\boldsymbol{x}'$. This is illustrated graphically in Figure 4.4 where the dashed line represents the true Pareto front, the grey dots represent the current solutions in the archive, and the two black dots represent the current solution and the neighbouring solution, respectively.

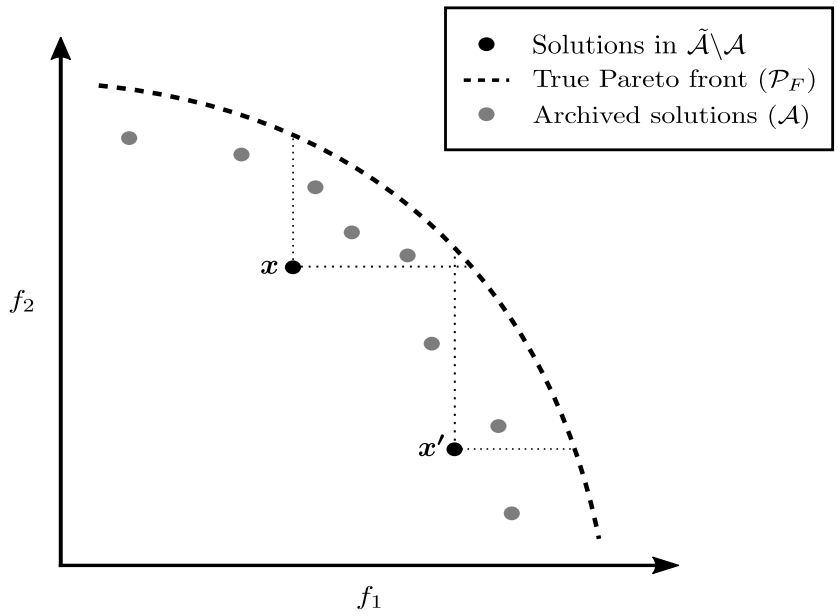


FIGURE 4.4: *The energy difference between the current solution $\boldsymbol{x}$ and neighbouring solution $\boldsymbol{x'}$ in a bi-objective optimisation problem in which both the objective functions, f_1 and f_2, are maximised.*

When considering the portion of the true Pareto front in Figure 4.4 where the current solution $\boldsymbol{x}$ and the neighbouring solution $\boldsymbol{x'}$ are dominated in objective space, it appears as though $\mu(\mathcal{P}_{x'}) > \mu(\mathcal{P}_x)$. This is, however, not the case since the number of solutions that dominates $\boldsymbol{x}$ in $\tilde{\mathcal{A}}$ is 3 and the number of solutions that dominates $\boldsymbol{x'}$ in $\tilde{\mathcal{A}}$ is 1, resulting in $|\tilde{\mathcal{A}}_{\boldsymbol{x}}| = 3 > 1 = |\tilde{\mathcal{A}}_{\boldsymbol{x'}}|$.

After a neighbouring solution $\boldsymbol{x'}$ has been generated from the neighbourhood of the current solution $\boldsymbol{x}$, the fitness value of $\boldsymbol{x'}$ is compared to the fitness value of $\boldsymbol{x}$. Again the *Metropolis acceptance rule* [73] is adopted, as discussed in §4.3.1, which states that the neighbouring solution $\boldsymbol{x'}$ may be accepted as the new current solution for the next iteration with some probability when the fitness value of $\boldsymbol{x'}$ is worse that the fitness value of $\boldsymbol{x}$. According to the Metropolis acceptance rule, the acceptance probability is calculated as $\exp\left(-\frac{\triangle_E(\boldsymbol{x'},\boldsymbol{x})}{T_i}\right)$ where T_i denotes the temperature for iteration i. However, if the fitness value of $\boldsymbol{x'}$ exceeds the fitness value of $\boldsymbol{x}$, the neighbouring solution $\boldsymbol{x'}$ is accepted as the new current solution — that is with probability 1. The acceptance probability may be summarised as

$$P(\boldsymbol{x'}) = \min\left\{1, \exp\left(-\frac{\triangle_E(\boldsymbol{x'},\boldsymbol{x})}{T_c}\right)\right\}. \tag{4.12}$$

It should be noted that if the neighbouring solution $\boldsymbol{x'}$ is dominated by fewer elements in the current estimated Pareto front in $\mathcal{A}$ than the current solution $\boldsymbol{x}$, the neighbouring solution $\boldsymbol{x'}$ is automatically accepted as the new current solution — it is an improving move. If, however, the energy difference between $\boldsymbol{x'}$ and $\boldsymbol{x}$ is large positive (that is when $\boldsymbol{x'}$ is dominated by more solutions in the Pareto front in $\mathcal{A}$ than $\boldsymbol{x}$) and the temperature T_i for iteration i is low, then the acceptance probability $P(\boldsymbol{x'})$ of accepting solution $\boldsymbol{x'}$ will be small. The acceptance probability will, therefore, remain unchanged if the objective functions were to be rescaled since it does not depend on the a *priori* weighting of objectives.

As in the method of simulated annealing, the DBMOSA is also executed iteratively in stages know as epochs, where the temperature remains constant during an epoch. As mentioned in §4.3.1, the length of an epoch is also determined according to the success of the algorithm

whereas an epoch is terminated when the temperature is *cooled* or *reheated*. Similar to the method of simulated annealing, reducing the temperature makes it harder for the algorithm to accept worsening solutions whereas increasing the temperature aims to make it easier to accept worsening solutions. In the former case, the temperature is reduced when too many neighbouring solutions are accepted whereas in the latter case, the temperature is increased when too little neighbouring solutions are accepted. The same cooling and reheating schedules described in §4.3.1 may be used in the DBMOSA.

A stopping criterion is also employed in the DBMOSA. This may be iterating the algorithm until a fixed number of iterations is executed, or when a fixed number of successive epochs have elapsed without accepting a single solution. Once the algorithm is terminated, the archive $\mathcal{A}$ is returned as output which contains the set of nondominated solutions for the problem instance — *i.e.* the set of approximately Pareto optimal solutions.

4.3.4 The nondominated sorting genetic algorithm II

The NSGA II was proposed by Agarwal *et al.* [28] in 2002, and aims to approximate good Pareto optimal solutions for a MOOP of the form (3.1)–(3.4). The NSGA II is merely an extension of the GA, and therefore, shares fundamental concepts with this algorithm, as mentioned in §4.3.2. However, the main difference between the conventional GA and the NSGA II lies in the way that fitness values are assigned to each solution and the consequent effect in which crossover operators are applied when selecting parent solutions.

The working of the NSGA II is provided in pseudocode form as Algorithm 4.4. The NSGA II requires as input a solution vector $\boldsymbol{z}$ that contains the objective function values for each objective, the size of the population of candidate solutions N that is maintained throughout the search, a probability for crossover p_c, a probability for mutation p_m, and the maximum number of generations G_{max} the algorithm should be executed for. The algorithm is initiated by generating an initial feasible population of candidate solutions $\mathcal{P}_0$ of size N randomly, where each solution in $\mathcal{P}_0$ is then ranked and sorted according to the FNSA [28] (discussed in §3.6). For each solution in $\mathcal{P}_0$ a dominance count d_i^c is then computed (that is the number of solutions that dominates solution i) as well as a subset S_i of solutions within the current population that dominates solution i.

Next, the solutions are sorted into different *nondominated fronts* according to its dominance count. All the solutions that achieved a dominance count of $d_i^c = 0$ are placed in the first nondominated front, denoted by $\mathcal{F}_1$, and assigned a rank value 1. For each solution i in $\mathcal{F}_1$, the algorithm iterates through each solution in S_i, and reduces the d_j^c-value (which is the dominance count for the j^{th} solution in S_i) by one. This ensures that the effect of solution i is discounted on solution j's dominance count. Next, all the solutions in the population involving a rank value 1 are removed, while the remaining solutions in the current population that now achieves a dominance count of $d_i^c = 0$ are placed in the next nondominated front for the algorithm to cycle through, denoted by $\mathcal{F}_2$. The solutions in $\mathcal{F}_2$ are then assigned a rank value 2. This procedure is iterated in this fashion until all the solutions in $\mathcal{P}_0$ have been assigned ranks.

Next, a *crowding distance* is estimated for each candidate solution in each respective nondominated front. This measure indicates the density of the solutions that surrounds the current solution, and typically takes on a high value when the solution is more isolated in objective space and a low value when the solution is less isolated in objective space. This measure is used to select solutions with a low density value for mating purposes in a bid to uncover solutions within regions of the objective space that have not been sufficiently explored. The calculation of

Algorithm 4.4: The NSGA II [28].

Input : An instance of a MOOP of the form (3.1)–(3.4), the population size N, the maximum number of generations G_{max}, the crossover probability p_c, and the mutation probability p_m

Output: A set of approximate Pareto optimal solutions $\mathcal{W}_{\mathcal{P}}$ to an instance of the MOOP of the form (3.1)–(3.4).

Generate random initial feasible population $\mathcal{P}_0$ of size N;
Rank and sort $\mathcal{P}_0$ using the FNSA (Algorithm 3.4);
Calculate the crowding distance for each solution in $\mathcal{P}_0$ using the crowding distance assignment algorithm (Algorithm 4.5);
Create an offspring population $\mathcal{Q}_0$ of size N using the binary tournament selection based on the crowding distance operator $\prec_{cco}$;
Perform crossover and mutation operators on selected solutions from $\mathcal{P}_0$;
$i \leftarrow 1$;
while $i \leq G_{max}$ **do**
    $\mathcal{R}_i \leftarrow \mathcal{P}_i \cup \mathcal{Q}_i$;
    Partition $\mathcal{R}_i$ into nondominated fronts $\mathcal{F}_1, \mathcal{F}_2, ...$ using the FNSA (Algorithm 3.4);
    $\mathcal{F}_{i+1} \leftarrow \emptyset$;
    $j \leftarrow 1$;
    while $|\mathcal{P}_{i+1}| < N$ **do**
        if $|\mathcal{F}_j| + |\mathcal{P}_{i+1}| \leq n$ **then**
            $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_{i+1} \cup \mathcal{F}_j$;
        else
            Calculate crowding distance for each solution $j \in \mathcal{F}_j$;
            Sort solutions in $\mathcal{F}$ in a descending order based on the crowding distance;
            $\mathcal{P}_{i+1} \leftarrow \mathcal{P}_{i+1} \cup \{$the first $(N - |\mathcal{P}_{i+1}|)$ solutions in $\mathcal{F}_j\}$;
        $j \leftarrow j + 1$;
    Calculate the crowding distance using FNSA (Algorithm 4.5) for each solution in $\mathcal{P}_{i+1}$;
    Create an offspring population $\mathcal{Q}_0$ of size N using the binary tournament selection based on the crowding distance operator $\prec_{cco}$;
    Perform crossover and mutation operators on selected solutions from $\mathcal{P}_0$;
    $i \leftarrow i + 1$;
return $\mathcal{W}_{\mathcal{P}} = \mathcal{P}_{G_{max}}$;

the crowding distance for each candidate solution in a nondominated front is given in pseudocode form as Algorithm 4.5.

In order to estimate the crowding distance, each solution in the population has to be sorted in an ascending order of magnitude along each objective axis. Let the objective function value of the i^{th} candidate solution for the m^{th} objective function be denoted by $Z[i]|m$, and let the crowding distance for solution i^{th} for the m^{th} objective function be denoted by $i_{dist}|m$. Moreover, let t denote the number of solutions in the population. Next, an infinite crowding distance for the boundary solutions in the population, $Z[1]|m$ and $Z[t]|m$, are assigned in order to ensure that they are easily selectable for crossover. The crowding distances for the intermediate solutions i are incremented by the normalised distance between their closest neighbours, where the normalised distance is calculated as $\frac{Z[i+1]|m - Z[i-1]|m}{m_{max} - m_{min}}$. In this calculation, m_{max} and m_{min} represents the maximum and minimum values of the m^{th} objective function. For each objective,

Algorithm 4.5: The crowding distance assignment algorithm [28].

Input : A population $\mathcal{F}$ of candidate solutions $\mathcal{F}$ to a MOOP of the form (3.1)–(3.4).
Output: The crowding distance $\mathcal{F}[i]_{dist}$ for each solution in $\mathcal{F}$

$n \leftarrow |\mathcal{F}|$;
forall $i \in \mathcal{F}$ **do**
  $\mathcal{F}[i]_{dist} \leftarrow 0$;
forall M *objectives* **do**
  $\mathcal{F} \leftarrow \text{sort}(\mathcal{F}, m)$;
  $\mathcal{F}[1]_{dist}|m \leftarrow \infty$;
  $\mathcal{F}[n]_{dist}|m \leftarrow \infty$;
  for $i = 2 \rightarrow (n-1)$ **do**
    $\mathcal{F}[i]_{dist}|m \leftarrow \mathcal{F}[i]_{dist}|m + \frac{\mathcal{F}[i+1]|m - \mathcal{F}[i-1]|m}{m_{max} - m_{min}}$;
return $\mathcal{F}[i]_{dist}$ for each solution in $\mathcal{F}$;

the overall crowding distance is taken as the collective value of the crowding distances of the individual solutions. An example of estimating the crowding distance in objective space for solution i is graphically illustrated in Figure 4.5 where both objective functions are minimised.

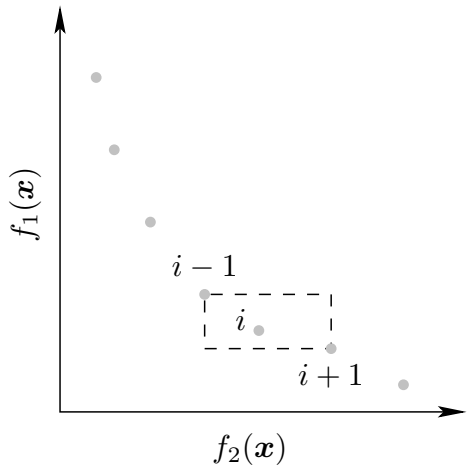


FIGURE 4.5: *Calculating the crowding distance for solution i in objective space, where a cuboid is formed around its two neighbouring solutions. For this bi-objective problem, objective functions $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ are to be minimised.*

Two selection criterion are employed in the NSGA II. The first criteria is the rank assigned to a solution. A solution that achieves a lower rank value is considered superior to a solution that achieves a higher rank value (that is, if rank $\boldsymbol{i} < \boldsymbol{j}$ then $\boldsymbol{i} \prec \boldsymbol{j}$). If the pair of solutions achieve the same rank values, then a *crowding distance comparison operator*, denoted by $\prec_{cco}$, may be employed as a second selection criteria. This entails incorporating the crowding distance measure. The solution that achieves the highest crowding distance is considered the superior solution (that is, if rank $\boldsymbol{i} = \boldsymbol{j}$ and $i_{dist}|m > j_{dist}|m$, then $\boldsymbol{i} \prec_{cco} \boldsymbol{j}$). In this way, the exploration of diverse solutions in less crowded regions of the objective space are encouraged, and may lead to a more uniform distributed Pareto front approximation [28].

The remainder of the working of the NSGA II is as follows. A population of candidate solutions are ranked and sorted iteratively into nondominated fronts using the FNSA. The crowding distances for each solution i in $\mathcal{P}_i$ are computed, and an offspring population $\mathcal{Q}_i$ is generated using a binary tournament selection procedure. Crossover and mutation operators on the selected parent solutions are then applied to generate offspring solutions. Next, a larger intermediate population of size $2N$ are formed, denoted by $\mathcal{R}_i$, which is the combination of the parent population $\mathcal{P}_i$ and the offspring population $\mathcal{Q}_i$. The intermediate population $\mathcal{R}_i$ are then ranked and sorted into nondominated fronts using the FNSA. The next population of candidate solutions $\mathcal{P}_{i+1}$ are then populated by including the solutions from the first nondominated front $\mathcal{F}_1$, and then from the second nondominated front $\mathcal{F}_2$, and so forth, until the size N of $\mathcal{P}_{i+1}$ has been reached. However, if all the solutions within a particular nondominated front cannot be added to $\mathcal{P}_{i+1}$, the solutions in the particular front are then sorted in a descending order of crowding distance and added to $\mathcal{P}_{i+1}$ starting with the solution achieving the largest crowding distance until the population size N has been reached. The process of generating the next population $\mathcal{P}_{i+1}$ of candidate solutions from the current population $\mathcal{P}_i$ in the NSGA II is graphically illustrated in Figure 4.6. After the initial population $\mathcal{P}_0$ has been created, the algorithm iterates in this fashion until a stopping criterion is reached. A popular stopping criterion is terminating the algorithm after a fixed number of iterations has been reached. Once the algorithm terminates, the set of approximate Pareto optimal solutions are returned as output.

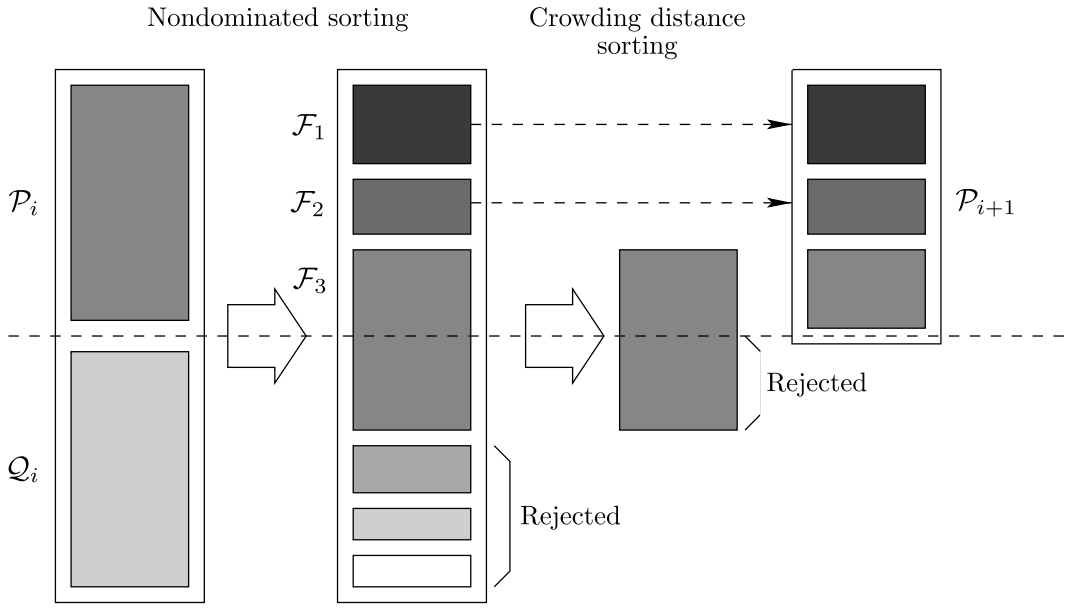


FIGURE 4.6: *A high-level schematic presentation of generating a new population $\mathcal{P}_{i+1}$ from the current population $\mathcal{P}_i$ in the NSGA II.*

4.4 Chapter summary

This chapter was dedicated towards a literature review on a number of solution approaches that may be used to solve single-objective optimisation problems and multi-objective optimisation problems. These solution approaches may be divided into three distinct groups. The chapter opened in §4.1 with a discussion on exact solution approaches, of which two exact methodologies

were briefly explained, namely the implicit branch-and-bound method proposed by Doig and Land [77] in 1970 and the explicit method of total enumeration.

Next, in §4.2, heuristic solution approaches were introduced and it was mentioned that these type of solution methodologies may be divided into three classes. An example of a heuristic solution method was also given in each of these classes. In §4.3, the focus shifted towards a discussion on metaheuristics. Four well-known metaheuristic solution methods were discussed, two of which may be employed to SOOPs and two of which may be employed to solve MOOPs. These methods included the method of simulated annealing proposed by Kirkpatrick *et al.* [73], the genetic algorithm proposed by Holland [63], an extension on the method of simulated annealing known as the DBMOSA and proposed by Smith *et al.* [115], and finally an extension on the genetic algorithm known as the NSGA II and proposed by Agarwal *et al.* [28]. The chapter finally closed in §4.4 with a brief summary of the chapter contents.

CHAPTER 5

Two novel mathematical models

Contents

In this chapter, two novel mathematical models are proposed which are able to suggest an irrigation schedule in conjunction with a scheme water supply schedule for farmers in order to manage water more effectively when water supply is limited. To the best knowledge of the author, the two crop irrigation and scheme water supply mathematical models formulated in this chapter are novel and have not appeared elsewhere in the literature.

The chapter opens in §5.1 with a discussion on a number of assumptions on which the formulation of the two mathematical models are based on. Next, in §5.2, a framework that focusses on the working dynamics of an open-air irrigation reservoir (*i.e.* the inputs required and processes involved) are put forward. An important aspect to consider when modelling the dynamics of such a reservoir is the effect of precipitation on the formulated irrigation and scheme water schedules, and is discussed in §5.3. Two solution approaches related to the nature of rainfall and whether rainfall should be considered as inflow to the reservoir are also briefly discussed in §5.3.

In §5.4, a novel single-objective crop irrigation and scheme water supply model is put forward with the objective to maximise the profit gained from irrigating crops while proposing an irrigation and scheme water supply schedule. The proposed schedule is, however, subjected to an user specified end-period reservoir water capacity. Next, in §5.5, a novel bi-objective crop irrigation and scheme water supply model is formulated and discussed comprehensively. This model is based on the single-objective crop irrigation and scheme water supply model of §5.4, but a second objective is introduced which aims to simultaneously maximise the end-period reservoir water capacity. This model provides multiple trade-off solutions between the total profit and end-period reservoir water capacity. Furthermore, in §5.6, a method to calculate the periodic reservoir end-volume as well as the end-period reservoir water capacity are briefly discussed. The chapter then closes in §5.7 with a brief summary of the chapter contents.

5.1 Reservoir behavioural assumptions

In order to develop a mathematical model which incorporates the operation of an open-air irrigation reservoir, a number of assumptions with respect to the behavioural activities of the reservoir have to be made. The following seven assumptions are made with respect to the reservoir of a farmer.

1. *Reservoir activities.* The proposed mathematical model takes into account a single open-air reservoir that is used only for the purpose of irrigating the crops that are currently grown on the farm. Open-air reservoirs that are designed for other activities (*i.e.* the generation of hydro-electricity or the storage of drinking water) are not considered here. Moreover, zero water losses (*i.e.* due to pipe leaks, faulty pumps, *etc.*) are also assumed when pumping water to the respective crop outlet areas.

2. *Time discretisation.* The allocation of irrigation water as well as scheduling scheme water supply over the scheduling horizon may be discretised into a number of smaller time intervals, called a *time period*, while a specific time period consists ten days. The scheduling horizon is taken as input from the user. It is also assumed that all months contain thirty days and that there are twelve months during a hydrological year. Therefore, thirty-six time periods may fall within a hydrological year. Furthermore, the mathematical models proposed later take into account a number of parameters (*i.e.* crop growth stages, the month of growth, crop yield response factors, and crop irrigation requirements) computed by CropWat 8.0 for a ten day time period. It is therefore important that the nature of the parameters taken as input to the mathematical models should correspond with the nature of the computed parameters from CropWat 8.0. A shorter time period (*e.g.* a ten day time period) may result in a more accurate estimation compared to a longer time period (*e.g.* a thirty day time period). Considering the time interval discretisation, the nature of this model is regarded as discreet. In this thesis, the scheduling horizon does not exceed one hydrological year, and may start at any time period within a hydrological year as long as the total extent of the time periods do not exceed one hydrological year (*i.e.* thirty-six time periods).

3. *Evaporation rate.* For a specific time period, the evaporation rate that a reservoir may experience is considered to be directly proportional to the average exposed water surface area of a reservoir, and is, therefore, also a function of the average reservoir volume, as described in §2.3. For a short time period (*i.e.* days or weeks), a realistic assumption may be to take the proportionality constant as dependent on the historical meteorological conditions of the time interval in question (in this case a ten day time period). Historical meteorological data may be collected from the South African Department of Water Affairs and Sanitation [34] (*i.e.* the daily A-pan evaporation rates) database — this database is updated annually. These daily A-pan evaporation rates are taken at a number of reservoirs that exceed a certain minimum storage capacity within regions of South Africa and also at specific locations. ClimWat 2.0, on the other hand, may also be used to determine meteorological data for a specified location, as discussed in §2.4.6.

4. *Silting rate.* The silting rate may be defined as the tempo at which the water capacity of a reservoir is reduced due to the accumulation of sediments on the reservoir floor [40]. This tempo is dependent on the duration of the time period under consideration. Since the scheduling horizon in the models developed here is only one year, the silting rate is considered negligibly small and are not taken into account in the model formulation. For

scheduling horizons exceeding one year, a realistic assumption is to take into account the fluctuation in the water level of a reservoir as a result of sediment buildup. Furthermore, multiple studies have shown that sediment buildup results in water to be unusable for drinking and irrigation purposes, and it is possible that sediment buildup may already exist within a reservoir. The percentage of unusable water is, however, subject to reservoir characteristics and the current sediment buildup level (that may be unknown). In this thesis, it is assumed that 10% of the total capacity of a reservoir is unusable due to sediment build up.

5. *Seepage rate.* The seepage rate is described as the tempo of water loss through the reservoir floor into the earth [82]. Measuring of this kind of water loss is very difficult and it is, therefore, assumed to be included in the bottom 10% of unusable water in a reservoir. It may be, for example, that sediment buildup contributes to 9% of unusable water and that 1% of the total capacity is lost due to seepage.

6. *Demand.* It is assumed that three types of resources may be considered when computing crop water requirements. They include the following: (a) it is assumed that farmers know the crop water requirements of each crop grown on his/her farm, (b) evaporative data collected from soil moisture probes is used, as described in §2.5, and (c) implementing computed crop water requirements performed by CropWat 8.0 based on a specific location and meteorological data. The mathematical model should, therefore, incorporate a database where the respective crop water requirements may be changed according to the farmer's preferences.

7. *Conservation law.* It is assumed that a change in the reservoir water capacity during any given time period is equal to the net flow (including precipitation, scheme water supply, borehole provision, *etc.*) minus the losses of evaporation and the volume of water to be irrigated to crops. Predicting precipitation is not encouraged in this thesis, however, it is rather taken as input to the mathematical model *after* precipitation occurred in millimetres. The mathematical model should therefore be re-run for updated values of the current reservoir water capacity after precipitation has occurred.

5.2 The proposed modelling framework

The proposed modelling of an open-air irrigation reservoir considered in this thesis is depicted in Figure 5.1. In this figure, a single open-air reservoir is subjected to a number of activities that result in the variation of reservoir water capacity. These activities include precipitation, evaporation, scheme water supply, borehole provision, the seepage rate, the silting rate and crop irrigation. According to assumptions 4 and 5 described earlier, the silting and seepage rate accounts for 10% of the reservoir water capacity to be unusable whereas the remaining activities are taken as user-input parameters to the mathematical model.

The user-input parameters to the model may be partitioned into three groups of dependent parameters and three individual independent parameters. The three groups of dependent parameters include demand-related parameters, farming-related parameters, and the reservoir-related parameters. These parameters are considered dependent since they are linked to one another in some sort of way (*e.g.* the amount of yield obtained by crops are subjected to the amount of hectares planted, the crop selling price and the amount of water available for irrigation). The independent parameters, on the other hand, include the scheduling horizon, the target end-period reservoir water capacity and the historical daily A-pan evaporation data. This historical

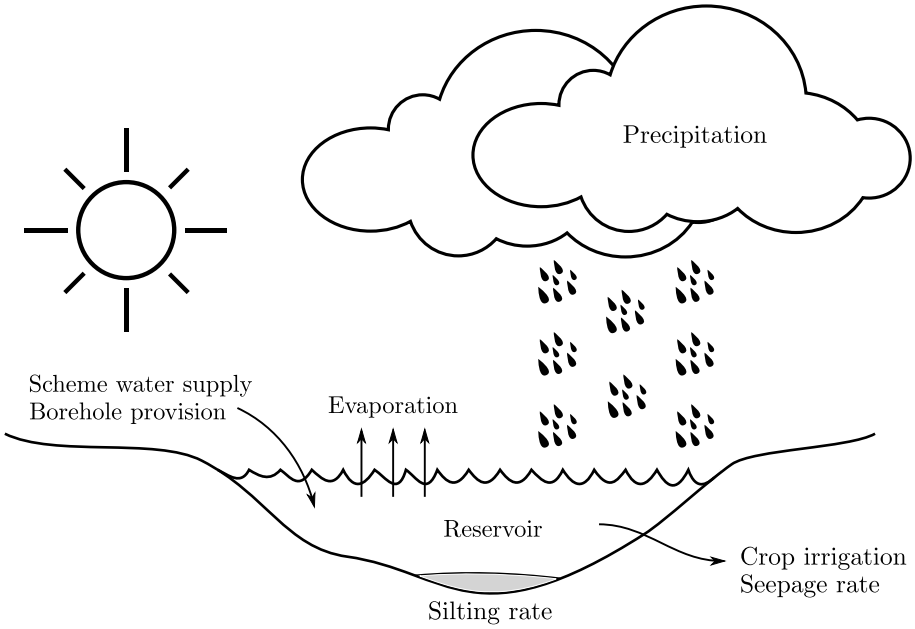


FIGURE 5.1: *A number of activities which affect the level of an open-air reservoir on a daily basis in a given hydrological year.*

A-pan evaporation data is used to calculate the coefficient of proportionality when estimating the evaporation losses from water surface areas.

The demand-related parameters include the crop water requirements for crops, the yield response factors (*i.e.* K_y) associated with each crop growth stage, the time and the type of crop growth stage that takes place in the hydrological year. Farming-related parameters refer to all the parameters that are required to estimate the profit gained from irrigating crops, and include the number of crops currently grown on the farm, the number of hectares planted for each crop type, the historical average yield achieved by a respective crop type, the selling price per ton for each crop, the variable cost for each crop, the fixed cost for each crop and the cost of obtaining additional water resources. The reservoir-related parameters include the holding capacity of the reservoir, the initial reservoir volume, the shape characteristic of the reservoir, the maximum amount of water that may be bought from scheme water supply, and the maximum and minimum amounts of water that may be acquired during a time period. As mentioned in §2.3, the shape characteristic of a reservoir relates the exposed water surface area of the reservoir with reservoir volume, and is also provided as an input to the model.

Given that the reservoir shape characteristic is known[1], the water surface area at any given volume may be calculated by applying a piecewise linear function to a sample of shape characteristic points. Moreover, a polynomial function is fitted to the historical A-pan evaporative data, and yields a smoothed-out evaporation rate curve for a specific location for every day in the hydrological year. It is then possible to calculate the evaporation losses for a time period as well as to estimate the new reservoir water capacity by incorporating the shape characteristic and the rate of evaporation. This approach was implemented by Van der Walt [134] in order to estimate the evaporation losses from the water surface area of the Keeromdam located in the Western Cape, South Africa. The aim of this study was to compute an effective water release strategy for local farmers.

[1]This assumption is considered reasonable since it is possible to determine a shape characteristic by means of a sonar scan [135].

Farms that are located next to river basins, are fortunate enough to have the option of making use of canal water or have boreholes on their farms to maintain a sufficient farm reservoir capacity for crop irrigation. Water may be pumped from the river basins into the reservoir or may be directed from a canal along an additional canal that feeds into the reservoir. This is typically known as scheme water supply and is usually measured by local Water Affairs by installing sensors on pumps or controlling the flow of water using sluices. If farmers have a borehole(s) on their farm, a steady flow is assumed that feeds into the reservoir. Finally, when considering rainfall that flows into the reservoir, a number of challenges are faced due to the variability of rainfall predictions. This calls for a careful investigation as to whether rainfall predictions should be considered as an inflow to reservoirs, and is thoroughly discussed in the next section.

5.3 Precipitation

As a result of global warming, multiple changes in the earth's climate have been witnessed over the past few decades and the magnitude of these events is becoming more extreme and they occur more frequently [89]. These events refer to heavy downpours, droughts, heat waves, hurricanes, and changing patterns in storms which make them difficult to predict. When considering precipitation and taking into account the ever changing weather patterns, the question arises whether yearly rainfall predictions are reliable or not. According to Epstein [44], a thirty day weather prediction is regarded worthless due to the variability of weather patterns. A shorter time window on which a weather prediction is made is considered more reliable with the reliability increasing as the prediction window becomes smaller. Therefore, Epstein [44] concludes that a three day forecast may be considered with 80% confidence as an accurate estimation for planned activities.

The expected yearly rainfall in a given region may be forecasted using a number of years of historical data. However, due to global warming, the probability exists that extreme weather conditions may occur (and regularly do). Furthermore, due to a very long time window, the variability of rainfall with respect to the quantity of rainfall and when to expect rainfall may vary highly. As an example of extreme weather conditions, consider the rainfall for the years 2015 and 2017 in the Western Cape province, South Africa. The low rainfall that occurred during these years may be seen as outliers compared to the rainfall experienced during other years, and resulted in the current drought this province is experiencing as mentioned by Piotr Wolski [146]. Rainfall predictions based on historical data may therefore be unable to predict such events, and the effect thereof on the local environment may be devastating.

The variability associated with rainfall predictions give rise to two different approaches that are considered in this thesis. The first approach is the prediction of rainfall as an inflow to an open-air reservoir whereas the second approach is based on the assumption that no rainfall will occur. The first approach is a more radical approach and entails predicting when rainfall will occur, the time span over which rainfall will occur, and to what extent rainfall occurs within this time span, and is typically achieved by using normal distributions, exponential distributions and geometric distributions. However, there is a risk associated with this approach since the probability exists that little to no rainfall may occur. The result of this may lead to overestimating reservoir water capacities on which the irrigation of crops and acquiring additional water resources are highly dependent on.

The second approach is considered a more conservative approach in the sense that the reservoir water capacity is predicted for the most extreme case (no rainfall). The available water resources may then be applied in a selective manner in order to irrigate crops with limited water resources.

Recall from §2.4.4 that it is possible to exploit crop growth stages that are sensitive to water deficits so that the actual yield is maximised. This may act as some sort of selective irrigation strategy with the aim to encourage efficient water use by maximising the crop yield when water supply is limited. Furthermore, additional water resources may then also be acquired to aid the current reservoir water capacity.

In conclusion, predicting yearly rainfall measures is not supported in this thesis due to the variability and risk associated with implementing the first approach, and therefore a more conservative approach is considered. To describe the devastating effect of inaccurate rainfall predictions on crop yield, a hypothetical scenario is introduced where a single crop is irrigated in growth stages with the aim to maximise crop yield and to obtain a specific reservoir water capacity. This scenario is based on the notions described in §2.3. Note that this scenario is specifically formulated to illustrate the risk associated with the aforementioned approaches.

Scenario. Suppose that a single crop is selected for planting over a period of four months while each month denotes a different crop growth stage (that is within the first month the first crop growth stage takes place, within the second month the second crop growth stage takes place, and so forth). Also, suppose that no additional water resources are available during the four months. For the first three months, the yield response factors are considered to be equal whereas the highest yield response factor takes place during the fourth month. The crop water requirements for the first three months are also equal, whereas the crop water requirement for the fourth month is double the amount of the first three months. An open-air reservoir is used to irrigate the crop of which a user-specified end-period reservoir water capacity of 20% should be reached at the end of month four. The aim is to find an irrigation strategy that irrigates crops in growth stages in order to maximise crop yield while obtaining a specified end-period reservoir water capacity. The first case represents a solution for the first approach whereas the second case represents a solution for the second approach.

Case 1. Consider that historical rainfall data was used to predict rainfall measures, and that these measures were taken as input to the reservoir. Taking into account the contribution of rainfall to the reservoir water capacity, it yields that sufficient water resources are available to meet crop water requirements. As a result, an irrigation strategy may now propose that 100% irrigation should be applied in each crop growth stage which results in a 21% reservoir water capacity at the end of month four. At the beginning of month four, it is noted that no rainfall has occurred since the start of the first month, and that the current reservoir water capacity is insufficient for supplying 100% irrigation to the fourth crop growth stage. Two solutions may then follow *i.e.* either no water is irrigated in order to ensure the specified end-period reservoir water capacity is met, or all the available water is irrigated in order to ensure minimum crop yield loss. When considering the prior solution, a large crop yield loss may be expected due to a water deficit taking place during the most sensitive crop growth stage.

Case 2. Now consider that rainfall is assumed and that the current reservoir capacity is insufficient for supplying 100% irrigation in all four crop growth stages. An irrigation strategy may now propose that 100% irrigation should be assigned to the fourth crop growth stage since this is the growth stage with the highest crop yield response factor, and irrigation may be lowered during the first three crop growth stages, respectively. At the beginning of the fourth month it is noted that no rainfall has occurred since the start of the first month. The fourth crop growth stage was therefore exploited in order to minimise the crop yield losses while the remaining irrigation water was divided among the first three crop growth stages, respectively. There is a small possibility that the crop yield loss in this case is more compared to case 1, however, a reservoir end-volume has been met at the end of month four.

Considering the two cases above, it is clear that the second case carries less risk in terms of the total crop yield loss and total profit. It is, therefore, incorporated in this thesis given the outcomes of the hypothetical scenario described above.

5.4 A single-objective crop irrigation and scheme water supply model

In order to formulate a single-objective crop irrigation and scheme water supply model, consider a farming environment which is similar to the reservoir modelling framework in Figure 5.1 (*i.e.* a farm that uses a single open-air irrigation reservoir to irrigate crops and may acquire additional water resources from scheme water supply). Let $j = \{1, ..., J\}$ denote a crop type, let $t = \{1, ..., T\}$ denote a specific growth stage during the life cycle of a crop, and let $p = \{1, ..., P\}$ denote a specific time period within the scheduling horizon. A binary parameter is also incorporated and acts as a linking parameter between crop growth and time of growth. The parameter is

$$y_{jp} = \begin{cases} 1 & \text{if a crop } j \text{ is grown during time period } p, \\ 0 & \text{otherwise.} \end{cases}$$

For calculation purposes, it is assumed that a crop growth stage is confined to a time period, and within a time period a crop growth stage for multiple crops may take place. Since a time period consists of such a short span (*i.e.* ten day time period), it is rarely the case that crop growth stages do not correspond with time periods as computed by CropWat 8.0.

In order to estimate the actual yield for a specific crop when the crop is subjected to a water deficit in one or more crop growth stages, a method that combines the approach taken by the FAO Irrigation and Drainage Paper No. 66 [118] and the multiplicative-type approach proposed by Stewart *et al.* [121] may be used. For this method, let Y_j^{actual} denote the actual yield gained (in tons per hectare) for crop j, let Y_j^{avg} denote the historical average yield obtained (in tons per hectare) for crop j, and let K_{jt} denote the yield response factor for crop j during growth stage t. Finally, let $\left(\frac{w}{ET}\right)_{jt}$ denote the decision variable where w is the amount of water irrigated to crop j during growth stage t and ET (total evapotranspiration) is the crop water requirement to crop j during growth stage t. The actual yield may then be estimated as

$$Y_j^{actual} = Y_j^{avg} \times \prod_{t \in T} \left(1 - K_{jt}\left(1 - \left(\frac{w}{ET}\right)_{jt}\right)\right). \tag{5.1}$$

The actual yield for crop j is calculated by multiplying the historical average yield obtained of crop j with a reduction fraction that relates inadequate water supply in a crop growth stage with a yield response factor assigned to a specific crop growth stage. The total reduction fraction is calculated by combining the reduced irrigation related to the crop water requirements in a crop growth stage t using a multiplicative method. Note that the actual yield is thus a function of the decision variable $\left(\frac{w}{ET}\right)_{jt}$.

Considering the decision variable $\left(\frac{w}{ET}\right)_{jt}$, there exists two possible outcomes. The first is the case where $w < ET$ for any $j \in J$ and $t \in T$. This indicates that the water irrigated w to crop j during growth stage t are less than the crop water requirement ET for crop j in growth stage t. This results in a decline in the actual yield since less water is irrigated to the crop. The

second is the case where $w = ET$ for any $j \in J$ and $t \in T$. This indicates that the crop water requirement for crop j during growth stage t is met. If this is the case, then $\left(\frac{w}{ET}\right)_{jt} = 1$ for $t = \{1, ..., 4\}$ which results in

$$Y_j^{actual} = Y_j^{avg}$$

since

$$\prod_{t \in T} \left(1 - K_{jt}\left(1 - \left(\frac{w}{ET}\right)_{jt}\right)\right) = 1, \;\; t \in T.$$

Some may argue that the case where $w > ET$ may follow which explains that more water is irrigated to the crop than what is required by the crop. However, this will not be considered since such a strategy is fundamentally logic.

During drought conditions, there exist a possibility that the current water capacity of the irrigation reservoir is insufficient to irrigate crops. Farmers may acquire additional water resources to aid the current reservoir water capacity, however, usually comes at a cost. Furthermore, it is also found that acquiring additional water resources are less expensive when compared to reducing irrigation to crops. For the mathematical model formulated here, acquiring of additional water resources by means of scheme water supply is modelled as two decision variables. The first is a binary decision variable denoted by Z_p and takes on a value 1 if scheme water supply is scheduled during time period p, or the value 0 otherwise. The second decision variable is denoted by O_p which indicates the amount of scheme water supply that should be acquired during time period p. If there is still insufficient water resources after the maximum amount of additional water resources has been obtained, the amount of water irrigated to crops may then be reduced.

Acquiring scheme water supply is limited in a number of ways, and entails that farmers are restricted to a maximum amount of scheme water supply for the length of the scheduling horizon due to limited available water resources for the season as well as ensuring that farmers downstream may benefit from scheme water supply. Farmers are also restricted to a maximum amount of scheme water supply during a respective time period p since only a specific amount of water may be pumped from rivers or directed from canals due to a limited pump speed or water flow. Furthermore, a minimum amount of scheme water supply is also incorporated in the model in order to ensure that the algorithm used to solve the model does not schedule scheme water supply that falls below the minimum — it is considered impractical to schedule scheme water supply for such low amounts of water. An alternative would be to add scheme water supply to periods where additional water is already obtained. Therefore, let $O_{p_{upper}}$ and $O_{p_{lower}}$ denote the maximum and minimum amount of scheme water that may be scheduled during a specific time period, respectively, and let O_{total} denote the maximum amount of scheme water that may be scheduled over the scheduling horizon. Furthermore, let O_{price} denote the cost of buying one m^3 of additional water resources using a scheme water supply.

When scheduling scheme water supply during time periods where evaporation rates are high, an increase in water losses due to evaporation are experienced when compared to scheduling scheme water supply during time periods where evaporation rates are low. Scheme water supply may also be scheduled in such a way that crop growth stages with high yield response factors are exploited during times where evaporation rates are high. It is, therefore, possible to schedule scheme water supply in such a way that crop yield production is maximised while simultaneously considering the water losses to evaporation in order to promote effective water management.

The objective of the single-objective crop irrigation and scheme water supply model, therefore, aims to maximise the total profit gained from crop yield. The total profit is estimated by subtracting the variable cost for crop j, the fixed cost for crop j and the cost of obtaining additional water resources from the revenue generated by selling yield from crop j, of which the profit is then accumulated for all crops. In order to calculate the profit, the following parameters are required. Let l_j denote the number of hectares planted for crop j, let p_j denote the price per ton (in Rands per ton) gained from selling crop type j, let v_j denote the variable cost per hectare (in Rands per hectare) for crop j, and let d_j denote the fixed cost (in Rands) for crop j. Moreover, let V_p denote the reservoir water capacity at the end of time period p (where $p \neq P$), and let R_{cap} denote the total water holding capacity of the reservoir. Finally, let $V_{P_{end}}$ denote the reservoir water capacity at the end of the scheduling horizon P, and let $U_{P_{end}}$ denote a user specified end-period reservoir water capacity representing the amount of water that should be present in the reservoir at the end of the scheduling horizon P. The objective function in the single-objective crop irrigation and scheme water supply model is therefore to

$$\text{maximise} \quad \sum_{p\in P}\sum_{j\in J} y_{jp}\left(Y_j^{actual}\left(\left(\frac{w}{ET}\right)_{jt}\right) l_j p_j - v_j l_j - d_j - O_p\, O_{price}\right), \tag{5.2}$$

$$\text{subject to} \quad 0.1R_{cap} \leq V_p \leq R_{cap}, \qquad p \in P, \tag{5.3}$$

$$V_{P_{end}} \geq U_{P_{end}}, \tag{5.4}$$

$$O_{p_{lower}} \leq O_p \leq O_{p_{upper}}, \qquad p \in P, \tag{5.5}$$

$$\sum_{p\in P} O_p \leq O_{total}, \tag{5.6}$$

$$\sum_{p\in P} Z_p \leq P, \tag{5.7}$$

$$0.5 \leq \left(\frac{w}{ET}\right)_{jt} \leq 1, \qquad j \in J,\, t \in T, \tag{5.8}$$

$$\left(\frac{w}{ET}\right)_{j1} = 1, \qquad j \in J. \tag{5.9}$$

Constraint set (5.3) ensures that the reservoir water capacity during any time period p does not exceed the total capacity of the reservoir, or fall below 10% of the total capacity of the reservoir (as specified in assumption 5). Furthermore, constraint (5.4) ensures that the user-specified reservoir water capacity is met at the end of time period p while constraint set (5.5) ensures that scheduled scheme water supply during any time period p does not exceed a maximum allowable amount of water, and does not fall below a minimum amount of water. Moreover, constraint (5.6) ensures that accumulated scheduled scheme water supply does not exceed a maximum allowable amount of scheme water supply for the scheduling horizon P while constraint (5.7) ensures that the accumulated periods scheduled for scheme water supply do not exceed the total number of time periods in the scheduling horizon. Finally, constraint set (5.8) ensures that the decision variable does not assign excess water to crop j during growth stage t while constraint set (5.9) ensures that the crop water demand for the first crop growth stage is met for each crop.

Consider constraint set (5.8) where $\left(\frac{w}{ET}\right)_{jt} \geq 0.5$. According to the FAO Irrigation and Drainage paper No. 33 [35], the linear relationship between the actual yield and water irrigated may not hold if the amount of water irrigated to crops is less than half of the crop water requirement in each respective crop growth stage. As a result, equation (5.1) may not be utilised to predict the actual yield which is fundamental to the mathematical model presented above. Consider now

constraint set (5.9) where $\left(\frac{w}{ET}\right)_{j1} = 1$. As in the paper of Igbadun *et al.* [66], the lowering of irrigation in the first crop growth stage were purposely avoided in order to ensure effective crop establishment, and is also adopted in the mathematical model presented above.

5.5 A bi-objective crop irrigation and scheme water supply model

To some farmers it may be more beneficial to consider multiple reservoir water capacities at the end of the scheduling horizon (that may also be the reservoir capacity at the beginning of the next hydrological year) given that planning is to commence regarding water availability for planting different crops. A bi-objective crop irrigation and scheme water supply model may therefore be formulated which aims to maximise the total profit from crop yield while simultaneously maximises the reservoir capacity at the end of the scheduling horizon. In this way a number of solutions involving trade-offs between these objectives may be presented to a farmer as *decision support*. This may also provide decision support to the farmer with respect to the risk associated with different reservoir water capacities at the end of the scheduling horizon.

In order to formulate a bi-objective crop irrigation and scheme water supply model, consider the single-objective crop irrigation and scheme water supply model (5.2)–(5.9). The bi-objective model formulated here is therefore an adaptation of the single-objective model (5.2)–(5.9) and solved in the context of a multi-objective environment. Constraint (5.4) is not considered here since it is no longer required to obtain a specified reservoir capacity at the end of the scheduling horizon, and an additional constraint is added which ensures that the obtained profit does not result in a loss given that the solution is unimplementable to farmers. Following the same notations as in the single-objective crop irrigation and scheme water supply model (5.2)–(5.9), the bi-objective crop irrigation and scheme water supply model may be formulated as

$$\text{maximise} \quad \sum_{p\in P}\sum_{j\in J} y_{jp}\left(Y_j^{actual}\left(\left(\frac{w}{ET}\right)_{jt}\right) l_j p_j - v_j l_j - d_j - O_p\, O_{price}\right), \tag{5.10}$$

$$\text{maximise} \quad V_{P_{end}}\left(\left(\frac{w}{ET}\right)_{jt}, O_p\right), \tag{5.11}$$

$$\text{subject to} \quad 0.1R_{cap} \le V_p \le R_{cap}, \qquad p \in P, \tag{5.12}$$

$$O_{p_{lower}} \le O_p \le O_{p_{upper}}, \qquad p \in P, \tag{5.13}$$

$$\sum_{p\in P} O_p \le O_{total}, \tag{5.14}$$

$$\sum_{p\in P} Z_p \le P, \tag{5.15}$$

$$0.5 \le \left(\frac{w}{ET}\right)_{jt} \le 1, \qquad j \in J,\ t \in T, \tag{5.16}$$

$$\left(\frac{w}{ET}\right)_{j1} = 1, \qquad j \in J, \tag{5.17}$$

$$\sum_{p\in P}\sum_{j\in J} y_{jp}\left(Y_j^{actual}\left(\left(\frac{w}{ET}\right)_{jt}\right) l_j p_j - v_j l_j - d_j - O_p\, O_{price}\right) > 0. \tag{5.18}$$

The set of constraints (5.12)–(5.17) may be interpreted in the same way as constraints (5.3)–(5.9) in the single-objective crop irrigation and scheme water supply model (5.2)–(5.9). In addition, constraint (5.18) ensures that the objective function value of the model is positive in order to propose implementable solutions to farmers. From this bi-objective crop irrigation and scheme water supply model (5.10)–(5.18), a farmer may select a single solution from an approximated Pareto optimal set of solutions based on the risk he/she is able to undertake when irrigating crops with limited water resources.

5.6 Computing periodic reservoir end-volumes

Again consider the proposed modelling framework in Figure 5.1. Let B_p denote a constant borehole inflow during time period p, let O_p be as previously defined, and let R_p denote the amount of rainfall that occurred during time period p. The total flow I_p for time period p into the reservoir may then be computed as

$$I_p = B_p + O_p + R_p, \quad p \in P. \tag{5.19}$$

Moreover, let x_p denote the amount of water irrigated to crops during time period p, let W_{jt} denote the crop water requirements for crop j during growth stage t, and let y_{jp} and $\left(\frac{w}{ET}\right)_{jt}$ be as previously defined. The total amount of water irrigated to crops during time period p may then be calculated as

$$x_p = y_{jp} \times W_{jt} \times \left(\frac{w}{ET}\right)_{jt}, \quad p \in P. \tag{5.20}$$

Furthermore, let E_p denote the total evaporation taking place from the reservoir water surface area during time period p, let e_p denote the daily evaporation rate associated with time period p, and let A_p denote the exposed reservoir water surface area during time period p. According to the third assumption made in §5.1, the total evaporation may be calculated as

$$E_p = e_p \left(\frac{A_p + A_{p+1}}{2}\right), \quad p \in P. \tag{5.21}$$

The shape characteristic f relates the exposed surface area of the reservoir with the stored volume of water in the sense that

$$A_p = f\left(V_p\right), \quad p \in P, \tag{5.22}$$

as incorporated in equation (5.21). Then, according to the seventh assumption made in §5.1, it follows that the periodic reservoir end-volume may be calculated as

$$V_{p+1} = V_p + I_p - x_p - E_p, \quad p \in P. \tag{5.23}$$

It is clear that the periodic reservoir end-volume is thus affected by the allocated values to the decision variables $\left(\frac{w}{ET}\right)_{jt}$ and O_p. These decision variables may, therefore, be varied in order to obtain a specified end-period reservoir water capacity or to maximise the end-period reservoir water capacity $V_{P_{end}}$ (that is where $p = P$ for V_p).

5.7 Chapter summary

The aim of this chapter was to formulate two novel mathematical crop irrigation and scheme water supply models which are able to encourage effective water management when water resources are limited. The chapter opened in §5.1 with a discussion on a number of assumptions upon which the two mathematical models were formulated. In §5.2, the proposed modelling framework on which the two mathematical models were formulated was briefly described, and followed by a brief discussion on a number of parameters grouped according to their nature as input to the two mathematical models. In §5.3, two solution approaches were suggested with regards to precipitation and the prediction thereof. From §5.3, it was found that less risk was associated with assuming that no rainfall will occur when irrigation planning is done compared to predicting rainfall measures as input the two mathematical models.

Sections §5.4 and §5.5 formed the heart of this chapter since it is here where the single-objective crop irrigation and scheme water supply model and the bi-objective crop irrigation and scheme water supply model were thoroughly discussed, respectively. The mathematical models were able to propose an effective irrigation and scheme water supply schedule within an environment where water availability is limited. Finally, in §5.6, the notions that are used to compute the reservoir water capacity at the end of each time period p were briefly discussed. The chapter then finally closed in §5.7 with a brief summary of the chapter contents.

CHAPTER 6

Model implementation and validation

Contents

This chapter is devoted to illustrate the working of the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18) described in Chapter 5 by solving them in the context of a small hypothetical farm scenario. The obtained results when solving the aforementioned mathematical models are used for validation purposes of both the mathematical models, and is done according to three respective validation methods. This chapter opens in §6.1 with a brief description on the small hypothetical scenario. This scenario entails a number of parameters that is used as input to the aforementioned mathematical models while incorporating an open-air irrigation reservoir located on the farm for irrigation purposes. Next, in §6.2, as part of the algorithmic implementation, the way in which neighbouring solutions are generated as well as the solution encoding, reheating and cooling schedules, and estimating the initial temperature for the respective solution methodology are briefly described.

This is followed by a parameter evaluation in §6.3 conducted on the algorithmic parameters associated with the adopted solution methodologies. The Wilcoxon signed-rank test is employed to estimate whether a statistical significance exists between different sets of parameter values with the aim to obtain a good set of parameter values for best algorithmic results. Furthermore, in §6.4, the results obtained from solving the two mathematical models are used as validation for the respective mathematical models. Three different validation methods are employed to validate

the models, and include a face validation, a random benchmark validation, and consulting an expert in the field of crop irrigation and farming. Finally, the chapter closes in §6.5 with a brief summary of the chapter contents.

6.1 A small hypothetical farm scenario

In this section, a small hypothetical farming scenario is developed with the aim to solve the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18). The small hypothetical scenario mimics a real-life farming scenario and, therefore, is adopted later for validation purposes.

In South Africa, maize is considered as the most important grain crop and consists of two variants, namely white and yellow maize [31]. Of these two variants, white maize is primarily used for human consumption and accounts for 43% of the total maize production in South Africa, whereas yellow maize is primarily used for animal feed production and accounts for 57% of the total maize production in South Africa. The statistical results were recorded by the South African government in the year 2016. Furthermore, the Western Cape accounts for 1% of the total maize produced during year 2015–2016 — growing maize in the Western Cape is not that common compared to growing maize in Mpumalanga, where 30% of total maize production was produced in the year 2015–2016. However, there are multiple success stories related to the growing of maize in the Western Cape. One of these was the result of low milk prices during 2012 where farmers in the Swartland district decided to plant yellow maize in order to produce silage for cows. During this time period, producing animal feed was considered less expensive than buying animal feed from international or local markets, as a worldwide maize shortage resulted in an increased price per ton [45].

For the small hypothetical scenario, suppose that a farmer in the Bredasdorp region has center pivot irrigation and grows yellow maize in order to supply animal feed to the local market. An open-air reservoir is used to irrigate 200 hectare of maize. The farm is also located next to the Karsrivier and water rights have been granted to the farmer in order to pump additional water from the river into the reservoir, if required to do so. However, the local water affairs keeps track of how much water is pumped daily as well as the total amount of water pumped over the scheduling horizon by means of a water meter installed at the pump. Suppose further that multiple weather institutions predicted a dry year with respect to rainfall in the Bredasdorp region. As a result, water supply in the region is restricted and farmers may only pump $100\,000\,m^3$ of water in total over the scheduling horizon. Farmers are also limited to using at most $8\,000\,m^3$ of water during each time period and at least $1\,000\,m^3$ of water should be pumped during a time period (that is if additional scheme water supply is to be used). A farmer is limited to a maximum amount of scheme water supply per time in order to ensure that farmers downstream have sufficient water supply, where a minimum scheme water supply is assigned per time period for impractical reasons — such reasons being pumping water for an hour within a time period, for example. Finally, the time continuum taken into account is a hydrological year which is partitioned into 36 time periods (that is 10 days in a time period).

The production cost associated with growing maize in the Bredasdorp region may relate to some extent with production costs in multiple other areas and regions in South Africa. In other words, the fixed costs and variable costs may differ between regions across South Africa, however, within a tolerance of production costs for other regions. A production cost budget for maize may be found on the Grain SA [58] website. Furthermore, according to the local Water Affairs office, the cost of scheme water supply may vary between R 3.10 – R 4.20 per m^3 of water. In Table 6.1,

the production costs, water costs and water limitations are illustrated as considered in the small hypothetical scenario. The production cost were taken from the crop production budget on the Grain SA [58] website.

Parameter	Quantity
Number of crops	1
Number of hectares	200 ha
Selling price	R 2 150
Variable cost	R 6 133
Fixed cost	R 2 024
Water cost	R 3.90
Total pumping capacity	100 000 m^3
Maximum pumping capacity per time period	8 000 m^3
Minimum pumping capacity per time period	1 000 m^3

TABLE 6.1: *A summary of the input parameters for maize as implemented in the hypothetical farming scenario. These parameters include the number of crops, the number of hectares planted, the production costs, the water costs, and the scheme water supply restrictions.*

The demand-related parameters for yellow maize were gathered from CropWat 8.0, according to the nearest weather station, and are illustrated in Table 6.2. This data consists of the time period of growth, the corresponding month of growth, the crop growth stage associated with each month of growth, the crop water demands (ET_c) and the crop yield response factors (K_y) associated with each crop growth stage. From Table 6.2, note that the initial growth stage stretches over two time periods, the second growth stage stretches over four time periods, the third growth stage stretches over four time periods, and the last growth stage stretches over three time periods. Also, note that maize is planted in August and harvested in December — irrigation is therefore required in-between these periods. Moreover, the average yield for maize was taken as 9 tons per hectare (irrigated) according to historical data [39].

Time period	Month	Growth stage	ET_c	K_y
31	August	Initial	3.1 mm	0.4
32	August	Initial	9 mm	0.4
33	August	Flowering	8.4 mm	0.4
34	September	Flowering	7.3 mm	0.4
35	September	Flowering	6.3 mm	0.4
36	September	Flowering	7.7 mm	0.4
1	October	Yield formation	9.9 mm	1.3
2	October	Yield formation	11.3 mm	1.3
3	October	Yield formation	10.0 mm	1.3
4	November	Yield formation	8.5 mm	1.3
5	November	Ripening	7.5 mm	0.5
6	November	Ripening	6 mm	0.5
7	December	Ripening	4.2 mm	0.5

TABLE 6.2: *A summary of the demand-related parameters for maize as implemented in the hypothetical farming scenario. The parameters include the time period of growth, the corresponding month of growth, the crop growth stage, the crop water demands and the crop yield response factors.*

A hypothetical reservoir was developed based on the shape characteristic of the Keeromdam in the Western Cape. The reservoir capacity was reduced relative to exposed water surface area in order to obtain realistic shape characteristic data points — typically farmers tend to scale

the amount of crops that they produce according to the amount of water that is available, and *vice versa*. In other words, if the farmer would like to produce more crops demanding additional water resources leading to the building of an additional irrigation reservoir, the reservoir water capacity would then be determined by the additional crop water demands. The obtained shape characteristic of the hypothetical reservoir which relates the reservoir volume with the exposed water surface area is illustrated in Table 6.3. Moreover, given that a linear approximation regression is fitted to the data in Table 6.3, the result thereof is illustrated in Figure 6.1 where the orange dots relate the shape characteristic data. In Figure 6.1, the x-axis denotes the reservoir water volume whereas the y-axis denotes the exposed water surface area.

Volume (in m^3)	Surface area (in m^2)
868.93	274.32
1775.24	1578.97
5990.09	3622.63
13513.47	5498.13
24345.45	7256.50
31002.13	8107.71
38485.95	8949.04
46796.93	9787.07
55935.02	10627.58
65900.27	11477.69
76692.66	12343.16
88312.16	13230.85
100758.82	14147.34
114032.49	15098.39
128133.58	16099.38
143061.65	17131.36
158816.86	18225.88
175399.20	19381.30
192808.69	20603.38
211045.32	21899.26
230109.10	23274.68
250000.00	24736.51

TABLE 6.3: *The shape characteristic of the hypothetical reservoir which illustrates the reservoir water surface area (in m^2) for reservoir water volume (in m^3).*

From the shape characteristic of the reservoir in Table 6.3 it is clear that the holding capacity of the reservoir is 2 500 000 m^3. Moreover, it was expressed by the farmer that a reservoir water capacity of 50% is aimed for at the end of the scheduling horizon. In order to achieve this, large volumes of scheme water supply should be acquired. At the current time period, the farmer has an initial reservoir water capacity of 92%. Moreover, from Weather South Africa [142], the daily A-pan evaporation data for the year 1964–1965 until the year 2005–2006 for Bredasdorp were gathered. The historical average daily evaporation rate was then calculated from the historical data for each day in a year. In Figure 6.2, the daily average evaporation rate is illustrated along with a 7^{th} degree polynomial applied to the A-pan evaporation rates.

The polynomial function fitted to the evaporation rates was achieved by using a least squares regression. The degree of the polynomial fit was incrementally increased until the best fit was acquired. The corresponding least squares regression errors are shown in Table 6.4. Considering the R^2 errors as well as the visual fit in Figure 6.2, it appears that the coefficients for a 7^{th}

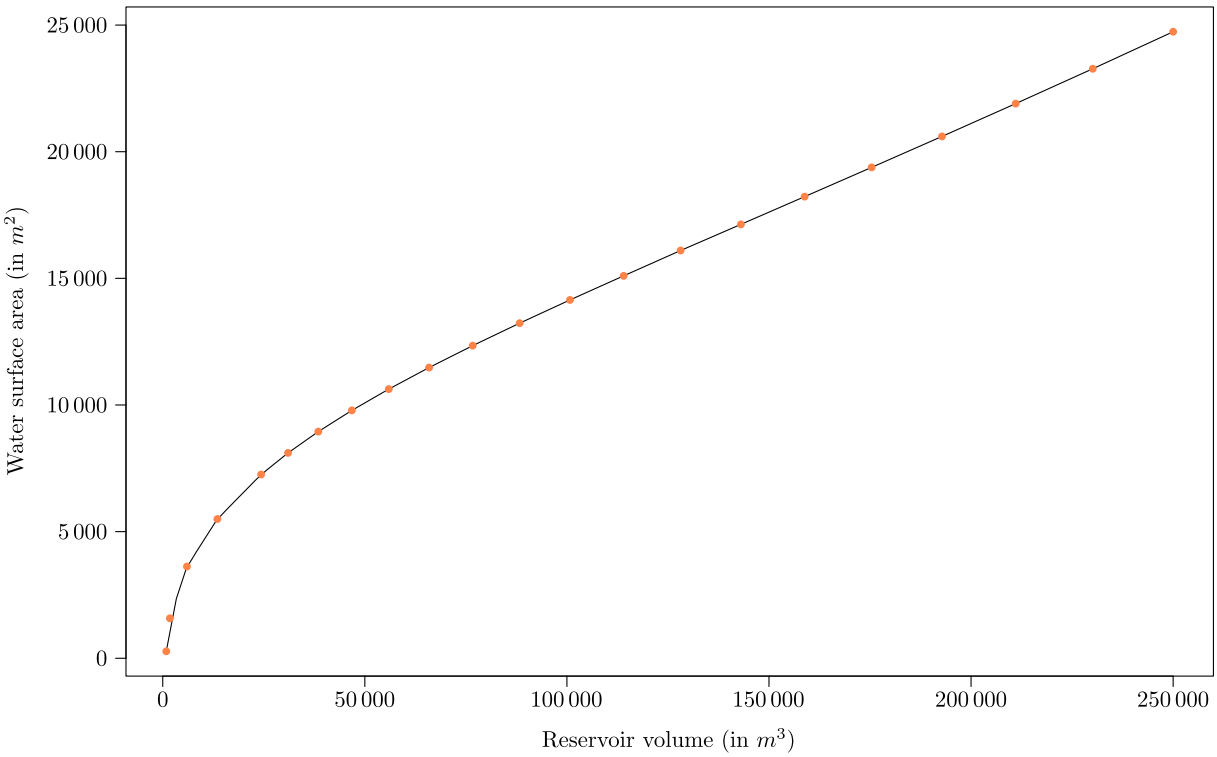


FIGURE 6.1: *A piecewise linear function relating the reservoir's volume (in m^3) with the exposed water surface area (in m^2) for the reservoir used in the hypothetical farm scenario.*

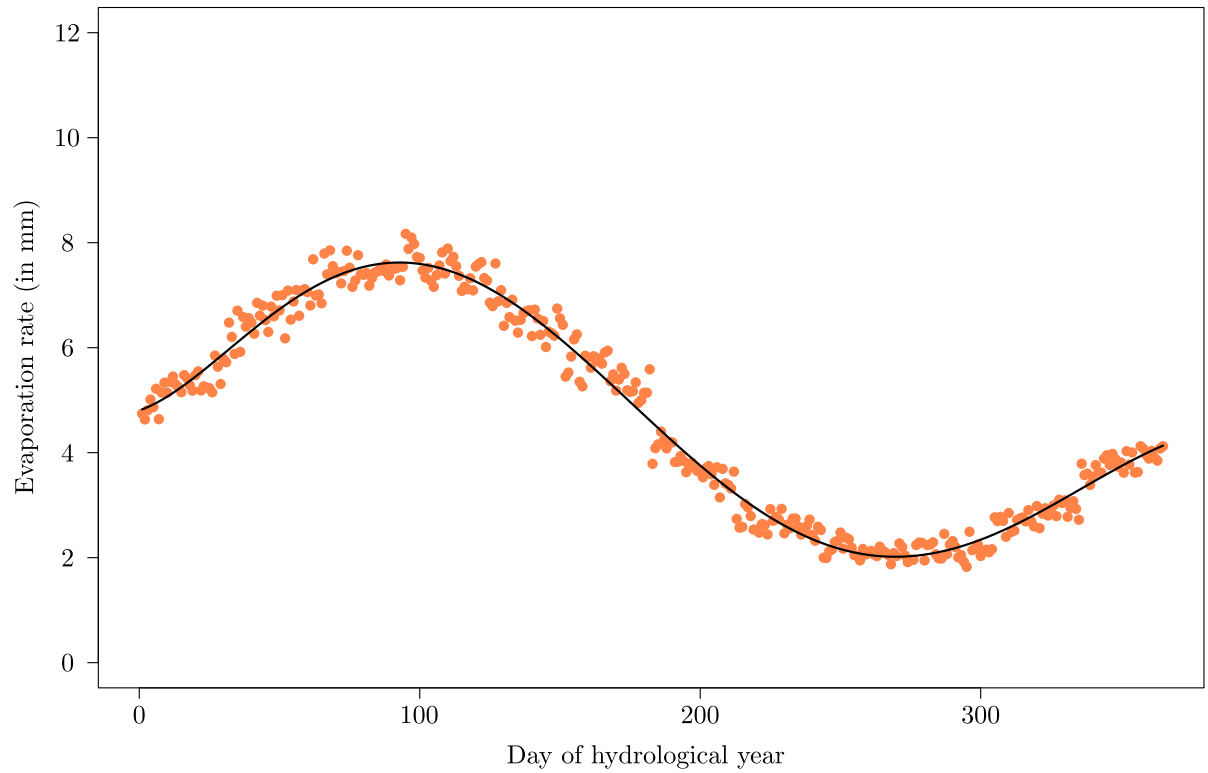


FIGURE 6.2: *A seventh-degree polynomial function fitted to the historical daily evaporation rate (in mm) experienced at Bredasdorp.*

degree polynomial fit are sufficient. According to the R^2 values in Table 6.4, it also appears that any degree of polynomial fit are sufficient due to the low error margins. The low R^2 errors for a 3^{rd} degree polynomial fit seem to be abnormal given that the fitted function is merely a sinus wave. Two computational packages were, therefore, used and included the `lm()` function that is preloaded in RStudio [108] for estimating the R^2 errors in Table 6.4, while the `polyfit()` function from the *pracma* package [8] (also in RStudio [108]) was used for validating the R^2 errors. The `lm()` function simply creates a simple regression model for the shape characteristic in Table 6.2 whereas the `polyfit()` function fits all polynomials up to the maximum specified degree. It was found that the R^2 values from both these functions correspond to the sixth decimal with one another.

Degree	R^2 errors (in %)
3	96.425%
5	98.287%
7	98.335%
10	98.386%
15	98.406%

TABLE 6.4: *The degree of the polynomial function fitted to the average historical evaporation rate and the R^2 error in % for the corresponding degree of polynomial fit.*

6.2 Algorithmic implementation

The single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18) may be solved in the context of the small hypothetical scenario formulated in §6.1 using the metaheuristic solution methodologies described in §4.3. The SA algorithm is chosen to solve the single-objective model (5.2)–(5.9) and the DBMOSA is chosen to solve the bi-objective model (5.10)–(5.18) from the metaheuristic solution methodologies discussed in §4.3. Furthermore, the software suite RStudio [108] is used as a programming platform to implement the selected solution methodologies.

From Chapter 4, the GA and the SA algorithm for single-objective optimisation, and the NSGA II and the DBMOSA for multi-objective optimisation were considered as possible solution methodologies for solving the mathematical models (5.2)–(5.9) and (5.10)–(5.18). The GA was eliminated as a solution method due to the fact that when generating neighbouring solutions by means of crossover operator, infeasible solutions may be produced which requires a repair mechanism to repair the solutions. This also accounts for the NSGA II since the crossover operator is also employed in this solution method. Upon employing the repairing mechanism, the computational time for generating feasible solutions may increase dramatically. Infeasible solutions may also be generated when generating a mating pool of solutions by means of the binary tournament selection procedure.

Furthermore, applying a trajectory-based solution approach ensures that the solution vector may be changed in such a way that restricted areas in the feasible region are exploited. The feasible regions of the two mathematical models presented in this thesis are restricted as a result of the set of constraints associated with both the mathematical models, and therefore motivates the implementation of a trajectory-based trajectory-based solution approach rather than a population-based solution approach. A trajectory-based solution approach may also contribute towards efficient constraint handling techniques. Moreover, a similar study was done by Georgiou *et al.* [54] where the optimal reservoir release strategy was computed in order

to irrigate crops using a soil water balance equation. Georgiou *et al.* [54] concluded that the use of the SA algorithm is sufficient for such kind of problems. In the remainder of this section, all the aspects confined to implementing a solution method to solve the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18) in the context of the small hypothetical scenario described in §6.1 are thoroughly discussed.

6.2.1 Solution encoding

According to the SA algorithm and the DBMOSA, a solution encoding of some sort is required when generating an initial feasible solution or a neighbouring solution. This yields a three-part solution vector where each distinct part in the solution vector relates to a different type of variable. An example of the solution encoding is shown in Figure 6.3 where the three different parts of the solution vector are clearly illustrated by means of square brackets.

$$\left[\left[\left(\frac{w}{ET}\right)_{11},\left(\frac{w}{ET}\right)_{12},...,\left(\frac{w}{ET}\right)_{jt}\right],\left[Z_1,Z_2,...,Z_p\right],\left[O_1,O_2,...,O_p\right]\right]$$

FIGURE 6.3: *The solution encoding of a solution vector which is taken as input to the SA algorithm and the DBMOSA.*

The first part of the vector contains the decision variable $\left(\frac{w}{ET}\right)_{jt}$ which denotes the ratio of the amount of water irrigated over the ratio of the water requirement for crop j during growth stage t, as described in §2.4.2, and the second part of the vector contains the auxiliary variable Z_p which denotes the time periods scheduled for scheme water supply where $p = 1, ..., P$, as described in §5.4. Finally, the third part of the vector contains the decision variable O_p which denotes the amount of scheme water supply scheduled during time period p where $p = 1, ..., P$, as described in §5.4.

The decision variables in the solution vector are iteratively changed with the aim to obtain a set of variables that maximise or minimise the objective function value(s). The output of the model is, therefore, a set of good decision variables contained in the solution vector. As part of a constraint handling technique, only feasible solutions are generated iteratively.

6.2.2 Generating an initial feasible solution

As mentioned in the previous section, an initial feasible solution is required in order to initiate the SA algorithm or the DBMOSA. An initial solution is generated by assigning random values to the decision variable $\left(\frac{w}{ET}\right)_{jt}$, assigning random time periods for scheme water supply to the decision variable Z_p, and allocate random amounts of scheme water supply to decision variable O_p which corresponds to the random generate time periods in Z_p.

It follows that the current solution should then be tested for feasibility, and is achieved by examining whether any of the constraints and constraint sets for the single-objective model (5.3)–(5.6) or the bi-objective model (5.12)–(5.18) are violated, depending on the model of which an initial feasible solution is generated. If it is found that one of the constraints is violated (*i.e.* when an infeasible solution is generated), a procedure is employed which randomly select entries in the solution vector and apply a perturbation to the selected entry until a feasible solution is achieved. Consider that one of the decision variables $\left(\frac{w}{ET}\right)_{jt}$ or O_p is selected, the selected entry is then incremented or decremented by a *step* value. In the case where the decision variable Z_p is selected, time periods are either added or removed from the decision variable vector. This procedure is repeated in an iterative fashion until a feasible solution is found.

6.2.3 Cooling schedules

The cooling schedule is responsible for iteratively reducing the system temperature in order to encourage the algorithm to converge towards an optimal solution in the objective space, as mentioned in §4.3.1. Multiple cooling schedules exist in the literature ranging in time efficiency and accuracy. For the the two solution methods implemented in this thesis, the widely used geometric cooling schedule is adopted as a cooling schedule due its simplicity and effectiveness over a wide range of problems [130].

The cooling parameter α may take on a number of values, where $0.8 \leq \alpha \leq 0.99$ according to Vigeh [138] or $\alpha > 0.83$ according to Abdullah [1]. The α value is initially selected as 0.85, however, a parameter evaluation on this value is also executed in order to find the value that will achieve the best algorithmic results. Details on the parameter evaluation will be provided later on in this chapter.

6.2.4 Reheating schedules

Reheating of the system temperature is applied during the execution of the algorithm when no more improving moves are accepted as the current solution. This is done to promote rapid exploration in the solution space by increasing the probability of accepting worsening solutions, as mentioned in §4.3.1. For the purpose of this thesis, the reheating schedule proposed by Abdullah [1] is adopted where the current temperature is set as the initial temperature T_0. As previously mentioned, the solution space is considered rather restricted, and therefore, the algorithm is encouraged to explore the solution space more by allowing it to accept worsening solutions more frequently.

6.2.5 Estimating the initial temperature

According to Busetti [15], the initial temperature is estimated such that 80% of all non-improving moves are accepted. The pseudocode for estimating the initial temperature T_0 according to equation (4.2) is given in Algorithm 6.1.

The average increase in energy of the objective function is calculated separately for both the objectives $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$ to the bi-objective model (5.10)–(5.18). This is achieved in steps 2–11 in Algorithm 6.1. Since the objective function of the single-objective model (5.2)–(5.9) is similar to the first objective $f_1(\boldsymbol{x})$ of the bi-objective model (5.10)–(5.18), the initial temperature T_0 may be computed simultaneously using a single algorithm. The initial temperature for the single-objective model (5.2)–(5.9) is, therefore, denoted by T_{0_s} (calculated in step 12) whereas the initial temperature for the bi-objective model (5.10)–(5.18) is, therefore, denoted by T_{0_b} (calculated in step 13). When estimating T_{0_b}, the maximum temperature value between the two objective functions f_1 and f_2 of the bi-objective model (5.10)–(5.18) is selected since the largest initial temperature encourages more exploration across a broader area of the solution space.

It was found that when incorporating Algorithm 6.1 to estimate the initial temperatures for the small hypothetical scenario described in §6.1, the initial temperature is estimated as $T_{0_s} = 500\,000$ for the single-objective model (5.2)–(5.9) and $T_{0_b} = 500\,000$ for the bi-objective model (5.10)–(5.18).

Algorithm 6.1: **Estimating the initial temperature T_0.**

Input : An initial solution vector $\boldsymbol{x}$, initial objective function values $f_1(\boldsymbol{x})$ and $f_2(\boldsymbol{x})$, and the length w of the random walk.

Output: The initial temperature T_{0_s} and T_{0_b}.

1 Initialise $s \leftarrow 0, t \leftarrow 0, \text{Increase}_{f_1} \leftarrow 0, \text{Increase}_{f_2} \leftarrow 0$;
2 **for** $i = 1$ *to* w **do**
3 *Generate a neighbouring solution* $\boldsymbol{x'}$;
4 $\triangle_{E_{f_1}}(\boldsymbol{x}, \boldsymbol{x'}) \leftarrow f_1(\boldsymbol{x}) - f_1(\boldsymbol{x'})$;
5 $\triangle_{E_{f_2}}(\boldsymbol{x}, \boldsymbol{x'}) \leftarrow f_2(\boldsymbol{x}) - f_2(\boldsymbol{x'})$;
6 **if** $\triangle_{E_{f_1}}(\boldsymbol{x}, \boldsymbol{x'}) > 0$ **then**
7 $\text{Increase}_{f_1} \leftarrow \text{Increase}_{f_1} + \triangle_{E_{f_1}}(\boldsymbol{x}, \boldsymbol{x'})$;
8 $s \leftarrow s + 1$;
9 **if** $\triangle_{E_{f_2}}(\boldsymbol{x}, \boldsymbol{x'}) > 0$ **then**
10 $\text{Increase}_{f_2} \leftarrow \text{Increase}_{f_2} + \triangle_{E_{f_2}}(\boldsymbol{x}, \boldsymbol{x'})$;
11 $t \leftarrow t + 1$;
12 $T_{0_s} \leftarrow \frac{-\text{Increase}_{f_1}}{s\ln(0.8)}$;
13 $T_{0_b} \leftarrow \max\left[\frac{-\text{Increase}_{f_1}}{s\ln(0.8)}, \frac{-\text{Increase}_{f_2}}{t\ln(0.8)}\right]$;

6.2.6 The neighbouring move operator

During each iteration of the algorithm, a neighbouring solution is generated by changing a single entry in one of the three distinct decision variables contained in the solution vector, respectively, in a bid to uncover an optimal solution vector that maximises or minimises the objective function(s). Two types of perturbation are applied to the solution vector, and include a small or a large perturbation. A small perturbation is applied during an epoch iteration (when the system temperature is kept constant) whereas a large perturbation is applied at the end of an epoch (that is when reheating takes place). A large perturbation enables the algorithm to escape from local optima and jump towards a new area in the solution space where a small perturbation explores the current area in the solution space.

Consider the case where a small perturbation is applied to the solution vector. Let *step* denote a fixed value by which decision variable $\left(\frac{w}{ET}\right)_{jt}$ is randomly incremented or decremented, let *wstep* denote a fixed value by which decision variable O_p is randomly incremented or decremented, and let $O_{sum} = \sum_{p=1}^{P} O_p$ which is the total accumulated scheme water supply of the current solution. Moreover, it is important to note that a small perturbation only entails changing decision variables $\left(\frac{w}{ET}\right)_{jt}$ and O_p. The procedure for generating a neighbouring solution by applying a small perturbation to the solution vector are given in pseudocode form as Algorithm 6.2.

A random value r in step 2 determines which decision variable in the solution vector should be perturbed. Consider that $r < 0.5$; in steps 3–9 in Algorithm 6.2, a random crop j and a crop growth stage t are selected and then randomly incremented and decremented by a *step* value. Now, let's suppose that $r \geq 0.5$, the random value r_r determines which type of perturbation is applied to the O_p decision variable. If the accumulated scheme water supply O_p are greater or equal than the maximum allowable amount of scheme water supply O_{total}, and $r_r < 0.5$, two entries in the O_p decision vector are randomly selected where the first is incremented and the second is decremented, both with the same *wstep* value. However, if $r_r \geq 0.5$ in step 14, a single entry is randomly selected in the O_p decision vector and decremented by a *wstep* value.

Algorithm 6.2: **Employing a small perturbation on the solution vector.**

Input : A feasible solution vector $\boldsymbol{x}$ of the form in Figure 6.3, an incremental *step* parameter, and an incremental *wstep* parameter.

Output: A feasible neighbouring solution vector $\boldsymbol{x}'$ of the form in Figure 6.3.

1 **while** *neighbouring solution is infeasible* **do**
2 $r = rand(0, 1)$;
3 **if** $r < 0.5$ **then**
4 Randomly select $j \in J,\ t \in T$;
5 $r_{vect} = rand(0, 1)$;
6 **if** $r_{vect} < 0.5$ **then**
7 $\left(\frac{w}{ET}\right)_{jt} = \left(\frac{w}{ET}\right)_{jt} + step$;
8 **else**
9 $\left(\frac{w}{ET}\right)_{jt} = \left(\frac{w}{ET}\right)_{jt} - step$;
10 *Test for feasibility*;
11 **else**
12 $r_r = rand(0, 1)$;
13 **if** $O_{sum} \geq O_{total}$ **then**
14 **if** $r_r < 0.5$ **then**
15 Randomly select $a \in P$ for $O_a > 0,\ b \in P$ for $O_b > 0,\ b \neq a$;
16 $O_a = O_a + wstep,\ O_b = O_b - wstep$;
17 **else**
18 Randomly select $p \in P$ for $Z_p > 0$;
19 $O_p = O_p - wstep$;
20 **else**
21 Randomly select $p \in P$ for $O_p > 0$;
22 **if** $r_r < 0.5$ **then**
23 $O_p = O_p + wstep$;
24 **else**
25 $O_p = O_p - wstep$;
26 *Test for feasibility*;

Moreover, if $r \geq 0.5$ in step 2 and $O_{sum} < O_{total}$, a random value r_r determines which type of perturbation is applied to the O_p decision variable. If $r_r < 0.5$ in step 22, O_p is incremented by a *wstep* value, or otherwise decremented by a *wstep* value.

The condition incorporated in step 13 where $O_{sum} \geq O_{total}$ prevents that the decision variable O_p is incremented since it will produce an infeasible solution. By employing this condition, feasible solutions are produced iteratively by suggesting avenues for perturbation. This may also allow for producing neighbouring solution in a faster time span. Furthermore, each neighbouring solution that was generated is tested for feasibility before exiting the while loop in step 1. If the solution is infeasible, the perturbation is reversed where a new entry is then selected within the same perturbation environment and iterated until a feasible solution is reached, or when a maximum number of attempts have been reached.

Consider that a large perturbation is applied to the solution vector. Let P_{sum} denote the number of time periods that are scheduled for scheme water supply for the current solution, let P_{min} denote the minimum number of time periods required in order for the solution vector

to be feasible, and let P_{max} denote the maximum number of time periods required in order for the solution vector to be feasible. The large perturbation entails changing any one of the decision variables in the solution vector, respectively — that is incrementing or decrementing decision variables $\left(\frac{w}{ET}\right)_{jt}$ and O_p, or adding or removing entries in the decision variable Z_p. The procedure for generating a neighbouring solution by employing a large perturbation to the solution vector is given in pseudocode form as Algorithm 6.3.

Algorithm 6.3: **Employing a large perturbation on the solution vector.**

Input : A feasible solution vector $\boldsymbol{x}$ of the form in Figure 6.3.
Output: A feasible neighbouring solution vector $\boldsymbol{x}'$ of the form in Figure 6.3.

```
1  while neighbouring solution is infeasible do
2      r = rand(0, 1);
3      if r < 0.5 then
4          if P_sum ≥ P_max then
5              Randomly select p ∈ P where Z_p > 0;
6              Z_p = 0, O_p = 0;
7          else if P_sum ≤ P_min then
8              Randomly select p ∈ P where Z_p = 0;
9              Z_p = p and O_p = rand(O_{p_lower}, O_{p_upper});
10         else
11             r_r = rand(0, 1);
12             if r_r < 0.5 then
13                 Randomly select p ∈ P where Z_p = 0;
14                 Z_p = p and O_p = rand(O_{p_lower}, O_{p_upper});
15             else
16                 Randomly select p ∈ P where Z_p > 0;
17                 Z_p = 0, O_p = 0;
18         Test for feasibility;
19     else
20         Randomly select j ∈ J, t ∈ T;
21         step = rand(0.5, 1.0);
22         r_r = rand(0, 1);
23         if r_r < 0.5 then
24             (w/ET)_jt = (w/ET)_jt + step;
25         else
26             (w/ET)_jt = (w/ET)_jt − step;
27         Test for feasibility;
```

In Algorithm 6.3, a random value r in step 2 determines which decision variable in the solution vector should be perturbed. Suppose that $r < 0.5$; in steps 4–9, a time period of scheme water supply is removed when the maximum number of time periods P_{max} is scheduled for scheme water supply in the current solution, whereas a time period of scheme water supply is added if the minimum number of time periods P_{min} is schedule for scheme water supply in the current solution (*i.e.* that is when no time periods of scheme water supply is scheduled). However, if the two aforementioned conditions in steps 4 and 7 are not true, a random value r_r determines whether a time period of scheme water supply is added or removed from the schedule Z_p. If a

time period is added to the schedule (as in steps 9 or 14), a random amount of scheme water supply between $O_{p_{lower}}$ and $O_{p_{upper}}$ is assigned to O_p. If a time period is removed from the schedule, Z_p and O_p are simply taken as zero (as in steps 6 or 17). Consider now that $r \geq 0.5$; in steps 20–26, a random crop j and growth stage t are selected of which the selected entry is incremented or decremented according to a random generated *step* value between 0.5 and 1.0. In this case, the random value r_r determines whether the decision variable $\left(\frac{w}{ET}\right)_{jt}$ is incremented or decremented, as in steps 23 and 25. After each perturbation, the solution is tested for feasibility, and if it is found that the solution is infeasible, the perturbation is reversed. A new entry is then selected within the same perturbation environment and iterated until a feasible solution is reached, or when a maximum number of attempts has been reached.

In the case where no feasible solution is found within a specific perturbation environment and the maximum number of attempts has been reached, the neighbouring solution reverts back to the original solution before perturbation was applied, and is then accepted as the new current neighbouring solution. A maximum number of attempts is assigned to each perturbation environment in order to reduce the computational time of generating a neighbouring solution.

6.3 Algorithmic parameter evaluation

A number of parameters in the SA algorithm and the DBMOSA determines the success of the algorithm and may be adjusted for better algorithmic results. These parameters include the maximum number of iterations the algorithm executes (i_{max}), the length of an epoch iteration ($maxepoch$), the rate of cooling (α) adopted in the cooling schedule, the maximum number of moves accepted during an epoch before cooling ($maxaccept$), and the maximum number of moves attempted during an epoch before reheating ($maxattempt$). A parameter evaluation is conducted on the aforementioned algorithmic parameters in order to determine a set of parameter values that achieves the best algorithmic results.

Essentially, the way that neighbouring solutions are iteratively generated determines the success of the algorithm outcome whereas the algorithmic parameters are then chosen based on the quality of the algorithmic outcome. In this sense, the bi-objective model (5.10)–(5.18) is similar to the single-objective model (5.2)–(5.9) given that Algorithms 6.2 and 6.3, which are employed to generate neighbouring solutions, are incorporated in both the mathematical models (5.2)–(5.9) and (5.10)–(5.18). The well-known *Metropolis-Hastings rule* is also incorporated in both mathematical models (5.2)–(5.9) and (5.10)–(5.18), as discussed in §4.3.1. However, the probability of accepting worsening solutions differ between the two mathematical models and, therefore, the initial temperature T_0 is computed for each model, respectively. Regardless of this, these two mathematical models are similar in nature with respect to their working and the way in which neighbouring solutions are generated. Due to the similarity of the two mathematical models and the great computational expense associated with conducting another parameter evaluation on the bi-objective model (5.10)–(5.18), it was deem only necessary to conduct a parameter evaluation on the single-objective model (5.2)–(5.9).

In order to evaluate the results when solving the single-objective model (5.2)–(5.9) in the context of the hypothetical scenario for different sets of parameter values, the *Wilcoxon signed-rank test* [64] is employed. This test may be used to compare two matched samples (*i.e.* pairs of data points) in order to assess whether there is a significant difference between the medians of these two samples. This is done by converting the two samples into a single sample by taking the difference between each data point pair. The null hypothesis H_0 is that a sample has a median of zero whereas the alternative hypothesis H_1 is that the median of a sample is not zero [64].

Moreover, the p-value for each comparison between the two samples is computed and provides strong evidence of statistical significance between the two samples when the p-value ≤ 0.05.

Three assumptions are made in the Wilcoxon signed-rank test in order to determine whether a data set may be evaluated according to this method [76]. The three assumptions are that the dependent variable(s) should be measured at a continuous level, the independent variable(s) should contain a two categorical (related groups) and the distribution between the differences of the two related groups should be symmetrical in shape (that is no normal distribution is assumed) [76]. In evaluating the dependent and independent variable considered for the parameter evaluation, it is found that the Wilcoxon signed-rank test is a good choice for determining whether a statistical significance between multiple sets of parameter values exists.

At first, three values for each parameter are considered and grouped into a low, medium and large value. A sample consisting out of 30 data points is generated for each set of parameter values as done according to multiple statistical analysis [62]. The algorithm is, therefore, executed 30 times for the same set of parameter values of which the objective function value returned (a data point) for each repetition are captured within one sample. However, all possible combinations for the three values of the five parameters (a full factorial design experiment) are not considered but rather some combinations thereof, due to the great computational expense when solving for all possible combinations. The combinations considered when adopting a full factorial design result in $3^5 = 243$ experiments executed 30 times, resulting in a total of 7 290 times to execute the single-objective model (5.2)–(5.9) for different sets of parameter values. Each set of parameter values are numbered and called a scenario denoted by S. Each scenario that is employed in the parameter evaluation experiment is illustrated in Table 6.5. The average objective function value for the 30 solutions (data points) when incorporating a corresponding set of parameter values are also illustrated in this table, as well as the maximum obtained objective function found within the sample of 30 solutions.

S	i_{max}	*maxepoch*	α	*maxaccept*	*maxattempt*	Average	Maximum
1	**1000**	500	0.85	0.8	0.2	R 825 134.71	R 1 015 022.20
2	**2000**	500	0.85	0.8	0.2	R 855 613.25	R 1 171 578.76
3	**3000**	500	0.85	0.8	0.2	R 904 804.95	R 1 175 424.20
1	1000	**500**	0.85	0.8	0.2	R 825 134.71	R 1 015 022.20
4	1000	**1000**	0.85	0.8	0.2	R 857 915.38	R 1 206 476.76
5	1000	**1500**	0.85	0.8	0.2	R 877 875.27	R 1 216 299.40
1	1000	500	**0.85**	0.8	0.2	R 825 134.71	R 1 015 022.20
6	1000	500	**0.90**	0.8	0.2	R 823 744.68	R 1 053 566.67
7	1000	500	**0.95**	0.8	0.2	R 823 338.70	R 1 151 753.00
8	1000	500	0.85	**0.7**	0.2	R 771 233.83	R 1 049 837.00
1	1000	1000	0.85	**0.8**	0.2	R 825 134.71	R 1 015 022.20
9	1000	1500	0.85	**0.9**	0.2	R 807 655.14	R 1 194 393.83
10	1000	500	0.85	0.8	**0.1**	R 811 116.21	R 1 172 738.64
1	1000	500	0.90	0.8	**0.2**	R 825 134.71	R 1 015 022.20
11	1000	500	0.95	0.8	**0.3**	R 787 112.09	R 1 135 549.72

TABLE 6.5: *A number of sets of parameter values which is taken as the algorithmic parameters to solve the models for the small hypothetical scenario. The average objective function value for the 30 solutions as well as the maximum objective function value between the 30 solutions are given.*

The bold-faced entries in Table 6.5 denote a group where a specific parameter is varied while the remaining parameters are kept constant, and the comparisons between scenarios are then executed in the respective groups. It is also important to note that scenario 1 is taken as the

benchmark set of parameter values of which the corresponding group scenarios are tested against. Therefore, the fourth, seventh, eleventh, and fourteenth scenario are also called scenario 1.

The `wilcox.test()` function were used (an existing function in RStudio) in order to compute the p-values when comparing scenarios with each other. The resulting p-values are presented in Figure 6.4 where a colour scale is used to indicate a statistical significance — a darker blue block indicates a larger statistical significance as indicated by the legend on the right-hand side in Figure 6.4.

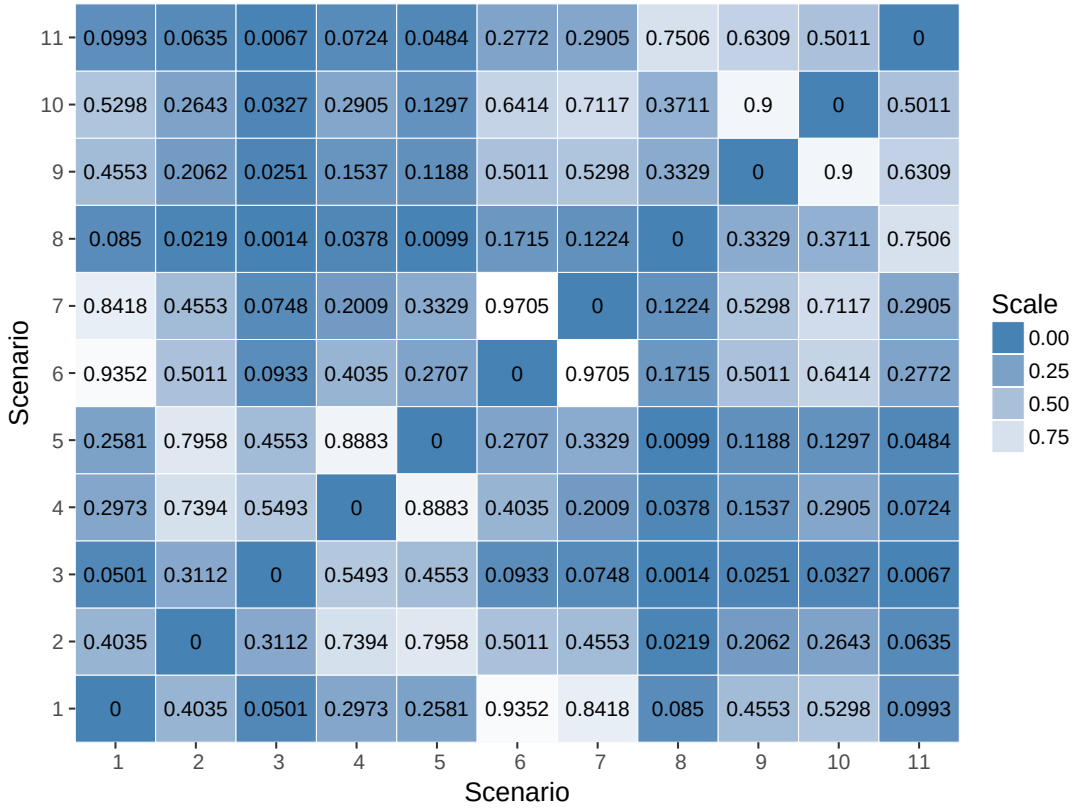


FIGURE 6.4: *The resulting p-values for scenarios 1–11 when employing the Wilcoxon signed-rank test.*

Consider scenarios 1, 2 and 3 where the number of i_{max} iterations are varied. From Figure 6.4, it is found that no evidence exists of a statistical significance between these scenarios since a p-value of 0.4035 was obtained for scenarios 1 and 2, a p-value of 0.0501 was obtained for scenarios 1 and 3, and a p-value of 0.3112 was obtained for scenarios 2 and 3. However, it is worth mentioning that a p-value of 0.0501 is obtained when comparing scenarios 1 and 3 which lies close to the level of significance of 0.05. Furthermore, when considering the average and maximum objective function values in Table 6.5 for scenarios 1 and 3, it is found that the average and maximum objective function values are notably larger for scenario 3 with an R 79 670.24 and an R 160 402 increase, respectively. This may be due to the larger i_{max} number of iterations involved in exploring a larger part of the solution space.

Moreover, consider scenarios 1, 4 and 5. It is also noted that no evidence exists of a statistical significance between these scenarios since a p-value of 0.2973 was obtained for scenarios 1 and 4, a p-value of 0.2581 was obtained for scenarios 1 and 5, and a p-value of 0.8883 was obtained for scenarios 4 and 5. Considering the average and maximum values in Table 6.5, a greater average and maximum objective function values are obtained for larger *maxepoch* iterations. This may be due to larger number of *maxepoch* iterations which explores the current local environment

within the solution space more, thus resulting in finding a local optimal solution. Considering the effect of i_{max} and *maxepoch* on the average and maximum objective function values, it is noted that a larger i_{max} number of iterations contributes towards greater average and maximum objective function values (that is an increase of R 79 670.24 and R 160 402 for an increased 2000 i_{max} iterations, respectively) whereas larger *maxepoch* iterations are more sensitive towards a greater maximum objective function value (that is an increase of R 201 277.20 for an increased 1000 *maxepoch* iterations). This view is supported by comparing scenario 3 with scenarios 4 and 5. It is found that scenario 3 resulted in a greater average objective function value of R 904 804.95 where scenarios 4 and 5 resulted in greater maximum objective function values of R 1 206 476.76 and R 1 216 299.40 respectively, for a 1000 i_{max} iterations.

Given that past runs of the algorithm using a random set of parameter values have resulted in a greater maximum objective function value when considering the maximum objective function values for scenarios 1–5, there may be some significance related to obtaining greater average and maximum objective function values when executing the algorithm for larger values of i_{max} and *maxepoch*. Consider now scenarios 1, 6 and 7 where the α parameter is varied. From the p-values in Figure 6.4, no evidence exists of a statistical significance between these scenarios since a p-value of 0.9352 was obtained for scenarios 1 and 6, a p-value of 0.8418 was obtained for scenarios 1 and 7, and a p-value of 0.9705 was obtained for scenarios 6 and 7. Conversely, the p-values for these scenarios are rather considered very large which indicates that there is similarity of sample medians between the different sets of parameter values when varying α. Moreover, considering the average objective function values, it is clear that for the different values of α the average objective function values remained in a close proximity to one another.

Consider now scenarios 8, 1 and 9 where the *maxaccept* are varied. From Figure 6.4, no evidence exists for a statistical significance between these scenarios since a p-value of 0.085 was obtained for scenarios 1 and 8, a p-value of 0.4553 was obtained for scenarios 1 and 9, and a p-value of 0.3329 was obtained for scenarios 8 and 9. When considering the average and maximum values in Table 6.5 for the corresponding scenarios, it is noted that the average objective function value obtained from scenario 8 is notably less than the average objective function values achieved for the other two scenarios. Furthermore, a large maximum objective function value has been obtained for scenario 9 of R 1 194 393.83 while a relatively low average objective function value of R 807 655.14 has been obtained. Taking the aforementioned into account, it seems that no correlation exists between different values of the *maxaccept* and the computed p-values, as well as the estimated average and maximum objective function values. The *maxaccept* is taken as 0.8.

Finally, consider scenarios 10, 1 and 11 where the *maxattempt* are varied. From the results in Figure 6.4, no evidence exists of a statistical significance between these scenarios since a p-value of 0.5298 was obtained for scenarios 1 and 10, a p-value of 0.0993 was obtained for scenarios 1 and 11, and a p-value of 0.5011 was obtained for scenarios 10 and 11. Considering the average and maximum values for the corresponding scenarios, it is found that scenario 11 achieved a notably lower average objective function value when compared to scenario 1, and scenario 10 achieved the largest maximum objective function value when compared to scenarios 1 and 11. Considering the aforementioned results, it seems that no correlation exists between different values of the *maxattempt* and the computed p-values, as well as the estimated average and maximum objective function values. The *maxattempt* is taken as 0.2.

The parameter evaluation experiment is also adopted for an extended i_{max} and *maxepoch* iterations. The new adopted values for i_{max} and *maxepoch* are shown in Table 6.6 where the benchmark set of parameter values are taken as scenario 12. From Table 6.6, the *maxepoch* parameter is executed for 2000 to 3500 iterations with a step of 500 iterations, and for the i_{max}

parameter for 1000 to 5000 iterations with a step of a 1000 iterations. However, the i_{max} iterations are executed for greater *maxepoch* values in order to exploit the sensitivity of i_{max} given that a larger *maxepoch* may promote a more stable environment in the sense that the objective function values may vary less. This is supported by the increasing average objective functions values for scenarios 12–15 for larger *maxepoch* iterations in Table 6.6. The resulting p-values for the extended parameter evaluation in Table 6.6 are illustrated in Figure 6.5, where again a darker blue block indicates a greater statistical significance between the scenarios.

S	i_{max}	*maxepoch*	α	*maxaccept*	*maxattempt*	Average	Maximum
12	1000	**2000**	0.85	0.8	0.2	R 858 126.80	R 1 124 685.40
13	1000	**2500**	0.85	0.8	0.2	R 902 377.68	R 1 200 144.67
14	1000	**3000**	0.85	0.8	0.2	R 942 375.43	R 1 257 135.00
15	1000	**3500**	0.85	0.8	0.2	R 959 937.16	R 1 202 481.24
12	**1000**	2000	0.85	0.8	0.2	R 858 126.80	R 1 124 685.40
16	**2000**	2000	0.85	0.8	0.2	R 901 273.49	R 1 210 079.64
17	**3000**	2000	0.85	0.8	0.2	R 900 613.08	R 1 253 257.00
18	**4000**	2000	0.85	0.8	0.2	R 971 250.11	R 1 257 525.00
19	**5000**	2000	0.85	0.8	0.2	R 984 181.58	R 1 255 870.36

TABLE 6.6: *An extended number of sets of parameter values which is taken as the algorithmic parameters to solve the small hypothetical scenario. The average objective function value for the 30 solutions as well as the maximum objective function value between the 30 solutions are also illustrated.*

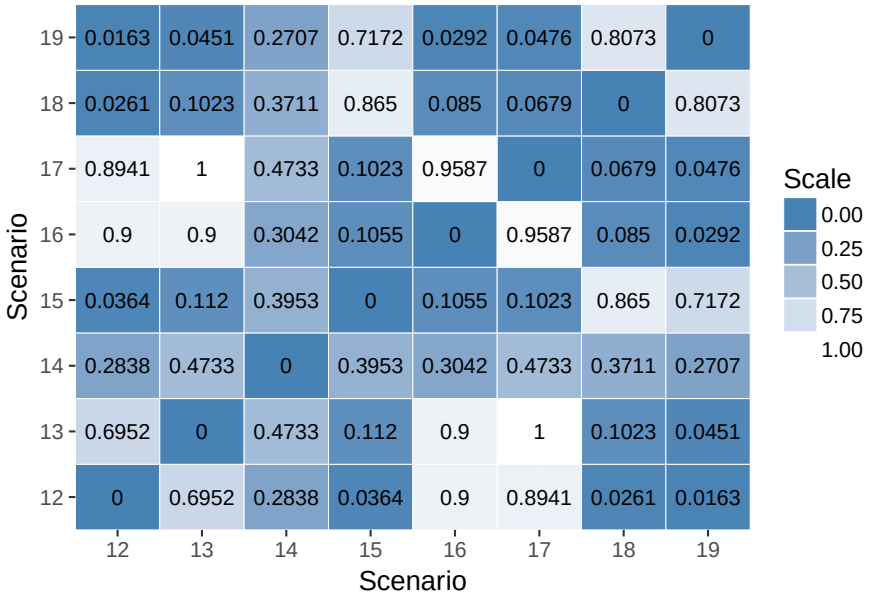


FIGURE 6.5: *The resulting p-values for scenarios 12–19 when employing the Wilcoxon signed-rank test.*

Considering scenarios 12–15 and the obtained p-values for the corresponding scenarios in Figure 6.5, there exist a strong evidence of statistical significance between scenarios 12 and 15 since a p-value of 0.0364 is obtained. This is also noted in the significant difference in the average and maximum objective function values between scenarios 12 and 15 which achieved a difference of R 101 810.36 and R 77 795.84, respectively. Moreover, it is also noted that scenario 14 achieved a greater maximum objective function value than scenario 15. This may be due to a small number of i_{max} iterations that was incorporated which may reduce the chance of obtaining an optimal solution if a smaller portion of the solution space are explored. Due to the

large computational time involved when incorporating large *maxepoch* iterations, the *maxepoch* parameter is only executed for a maximum of 3 500 iterations.

Considering scenarios 12, 16, 17, 18 and 19, strong evidence of statistical significance exists between scenarios 12 and 18, as well as scenarios 12 and 19 — the corresponding p-values yielded values of 0.0261 and 0.0163, respectively. The resulting p-value between scenarios 18 and 19 yielded 0.8073 which indicates a high similarity between these scenario medians. Considering scenarios 17 and 18, however, a p-value of 0.0679 is obtained which lies close to the level of significance, which is 0.005. Taking into account that larger i_{max} iterations result in an exponential growth of computational time to solve the problem at hand, it is proposed that the algorithm should be executed for a number of 4000 i_{max} iterations rather than 5000 i_{max} iterations, given that no statistical significance exists between scenarios 18 and 19 which achieved a p-value of 0.8073. Moreover, there is strong evidence of statistical significance between scenarios 16 and 19 and scenarios 17 and 19 since a p-value of 0.0292 and 0.0479 was achieved, respectively, which shows that no smaller than 4000 i_{max} iterations should be executed.

In conclusion, it was found that the α, *maxaccept* and *maxattempt* parameters do not play a significant role in the success of the algorithm results. Moreover, the results from the parameter evaluation proposed that 3 500 iterations for *maxepoch* parameter should be executed and 4000 iterations for the i_{max} parameter should be executed for best algorithmic results. Furthermore, the α, *maxaccept* and *maxattempt* parameters may be adjusted according to the decision maker, however, is initially taken as 0.85, 0.8 and 0.2, respectively. Moreover, in some cases the solution space of the single-objective model (5.2)–(5.9) is similar to the solution space of the bi-objective model (5.10)–(5.18) depending on the end-period reservoir water capacity $U_{P_{end}}$ set by the decision maker. The remaining constraints remains the same for the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18). For this reasons, above mentioned algorithmic results are also implemented for the bi-objective model (5.10)–(5.18).

6.4 Model validation

In this section, three methods are employed in order to validate the numerical results obtained when solving the single-objective model (5.2)–(5.9) in the context of the small hypothetical scenario formulated in §6.1. The first method is known as a face validation and involves a subjective evaluation of the realism and quality of a random feasible solution. The second method is a random benchmark validation and involves generating thirty random feasible solutions of which the solutions obtained by single-objective model (5.2)–(5.9) are compared to. Given that the objective function of the single-objective model (5.2)–(5.9) is also taken as an objective function in the bi-objective model (5.10)–(5.18), it is only necessary to compare a solution obtained from the prior model since fixing the single-objective model for multiple end-period reservoir water capacities result in a Pareto front when plotted. In essence, the bi-objective model (5.10)–(5.18) is then also validated. Finally, the third method involves consulting an expert in the field of crop irrigation and farming in order to validate the realism, authenticity and practicality of the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18), as well as the results obtained from solving these models.

6.4.1 A face validation

A face validation is carried out on an initial feasible solution in this section. Four key performance measures are computed for the initial feasible solution and include the total profit, the end-period

reservoir water capacity, the total cost of scheme water supply, and the total evaporation. The key performance measures are illustrated in Table 6.7 of which an total profit of R 563 422.90, an end-period reservoir water capacity of 131 654.50 m^3, a total cost of scheme water supply of R 363 090, and a total evaporation of 30 501.49 m^3 was obtained.

Total profit	R 563 422.90
End period reservoir volume	131 654.50 m^3
Total scheme water cost	R 363 090
Total evaporation	30 501.49 m^3

TABLE 6.7: *Four key performance measures when generating an initial feasible solution.*

Moreover, the resulting irrigation schedule for the initial feasible solution is illustrated in Table 6.8. From this table, the irrigation schedule proposes that 100% of the crop water requirements should be irrigated in the first growth stage for maize, 86% of the crop water requirements should be irrigated to second growth stage, 74% of the crop water requirements should be irrigated to third growth stage, 76% of the crop water requirements should be irrigated to fourth growth stage.

Crop growth stage	1	2	3	4
$\left(\frac{w}{ET}\right)_{jt}$	100%	86%	74%	76%

TABLE 6.8: *The irrigation schedule for the initial feasible random solution.*

Furthermore, the resulting scheme water supply schedule for the initial feasible solution is illustrated in Table 6.9. From this table, 17 time periods was assigned for scheme water supply and a total of 93 100 m^3 of water to be pumped during these time periods. The scheme water supply schedule suggest that 5 000 m^3 of water should be pumped during time period 11, 5 800 m^3 of water should be pumped during time period 12, 5 300 m^3 of water should be pumped during time period 13, 2 700 m^3 of water should be pumped during time period 15, 7 800 m^3 of water should be pumped during time period 18, 6 300 m^3 of water should be pumped during time period 19, 7 700 m^3 of water should be pumped during time period 20, 5 600 m^3 of water should be pumped during time period 22, 3 700 m^3 of water should be pumped during time period 24, 5 400 m^3 of water should be pumped during time period 25, 5 600 m^3 of water should be pumped during time period 28, 6 900 m^3 of water should be pumped during time period 29, 7 600 m^3 of water should be pumped during time period 31, 4 200 m^3 of water should be pumped during time period 32, 6 800 m^3 of water should be pumped during time period 33, 4 700 m^3 of water should be pumped during time period 34, and 2 000 m^3 of water should be pumped during time period 36.

Z_p	11	12	13	15	18	19	20	22	24
O_p	5 000	5 800	5 300	2 700	7 800	6 300	7 700	5 600	3 700
Z_p	25	28	29	31	32	33	34	36	
O_p	5 400	5 600	6 900	7 600	4 200	6 800	4 700	2 000	

TABLE 6.9: *The scheme water supply schedule for the initial feasible solution.*

The profit and the total cost of scheme water supply may by calculated by hand using the irrigation schedule in Table 6.8 and the scheme water supply schedule in Table 6.9, and then compared the this results with the results obtained in Table 6.7. Recall from equation (5.1) in §5.4, the actual yield may be calculated given that water deficits take place during the second, third and fourth crop growth stages. The Y_j^{avg} (in ton per hectare) are found in Table 6.1 while

the yield response factors K_{jt} associated with each crop growth stage are found in Table 6.2. The actual yield may then be calculated as

$$\begin{aligned} Y^j_{actual} &= 8.0 \times (1 - 0.4\,(1 - \mathbf{1}))\,(1 - 0.4\,(1 - \mathbf{0.86}))\,(1 - 1.3\,(1 - \mathbf{0.74}))\,(1 - 0.5\,(1 - \mathbf{0.76})) \\ &= 4.399 \text{ ton per ha,} \end{aligned}$$

where the values in bold denote the decision variables $\left(\frac{w}{ET}\right)_{jt}$ that are illustrated in Table 6.7. Next, the profit may be calculated by incorporating the input parameters (that is the number of hectares, the selling price, variable cost, *etc.*) in Table 6.1 as well as the calculated Y^j_{actual}. The total profit is estimated using equation (5.2) in §5.4 or equation (5.10) in §5.5. The resulting profit is then calculated as

$$\begin{aligned} \text{Total profit} = {} & \text{R}\,1\,520 \times 200 \text{ ha} \times 4.399 \text{ ton per ha} - (\text{R}\,2\,024 \text{ per ha} \times 200 \text{ ha}) - \text{R}\,6\,133 \\ & - \text{R}\,3.90 \times (5\,000 + 5\,800 + 5\,300 + 2\,700 + 7\,800 + 6\,300 + 7\,700 + 5\,600 + 3\,700) \\ & - \text{R}\,3.90 \times (5\,400 + 5\,600 + 6\,900 + 7\,600 + 4\,200 + 6\,800 + 4\,700 + 2\,000) \\ = {} & \text{R}\,563\,422.90 \end{aligned}$$

Considering the actual yield Y^j_{actual} and the total profit estimation above, it is noted that the total profit corresponds with the estimated total profit in Table 6.7, as computed by the single-objective model (5.2)–(5.9). Furthermore, considering the total scheme water supply scheduled during 17 time periods at a cost of R 3.90 per m^3, the total scheme water cost was estimated as R 363 090, which corresponds with the computed total scheme water cost in Table 6.7.

In order to determine the end-period reservoir water capacity and the total evaporation, the change in reservoir water capacity are computed in Excel using the shape characteristic data points (orange dots) in Figure 6.1, and the historical average daily evaporation rates in Figure 6.2. The calculation is executed in Excel since calculating the resulting end-period reservoir water capacity and total evaporation by hand is a large and complex task. It is important to note that a linear regression model was fitted through the shape characteristic data points whereas the historical daily average evaporation rates were used without fitting a polynomial function to it.

The resulting end-period reservoir water capacity and total evaporation are computed in Figure 6.6, and incorporates the irrigation schedule in Table 6.8 and the scheme water schedule in Table 6.9. In this figure, the outsource water relates to the borehole provision, rainfall and scheme water supply as defined in §5.6. In this case, borehole provision and rainfall were not considered as an inflow to the reservoir. Moreover, the demand (x_p) in Figure 6.6 was estimated according to equation (5.20) where the decision variables $\left(\frac{w}{ET}\right)_{jt}$ in Table 6.8 were used.

From Figure 6.6, the end-period reservoir water capacity and the total evaporation were calculated according to the notions described in §5.6. The appended red symbols in Figure 6.6 also correspond with the notions defined in §5.6. The resulting end-period reservoir water capacity is computed in row 55 in column E, and resulted in an estimated end-period reservoir water capacity of 131 856.982 m^3 of water. The total evaporation may be calculated by summarising the row values in rows 10, 21, 32, 43 and 54 which resulted in a total estimated evaporation of 30 299.018 m^3 over the scheduling horizon.

There exists no significant difference between the results obtained by the Excel model and the results obtained by the single-objective model (5.2)–(5.9) when generating an initial feasible

solution. However, the Excel model sparsely overestimated the end-period reservoir water capacity and the total evaporation, and may be due to the fact that the historical average daily evaporation rates were used rather than the coefficients of a fitted polynomial function to predict the average evaporation per time period.

6.4.2 A random benchmark validation

In this section, a random benchmark validation is carried out separately in this section for the single-objective model (5.2)–(5.9). The validation entails generating thirty random feasible solutions and then comparing the solution obtained by the single-objective model (5.2)–(5.9) with the results from the thirty random solutions. Furthermore, the aim of this validation is to see whether the solution obtained by the single-objective model (5.2)–(5.9) is better than the best solution among the thirty random generated solutions.

When solving the single-objective model (5.2)–(5.9) in the context of the small hypothetical scenario in §6.1 using the parameter values in §6.3, the following results were obtained. The solution resulted in a total profit of R 1 281 639, an end-period reservoir water capacity of 125 081.78 m^3 of water, a total scheme water cost of R 389 220, and a total evaporation of 27 702.22 m^3 of water. Furthermore, the irrigation schedule is illustrated in Table 6.10. This schedule proposes that irrigation should be reduced in the second crop growth stage by 36% of the crop water requirements whereas 100% of the crop water requirements should be irrigated during the remaining growth stages.

Crop growth stage	1	2	3	4
$\left(\frac{w}{ET}\right)_{jt}$	100%	64%	100%	100%

TABLE 6.10: *The irrigation schedule when solving the single-objective model (5.2)–(5.9) in the context of the small hypothetical scenario in §6.1 by employing the parameter values proposed in §6.3.*

The scheme water supply schedule proposed by the model yielded a total of 16 time periods of pumping water and a total of 99 800 m^3 of water to be pumped during these time periods. The obtained scheme water supply schedule containing Z_p and O_p are illustrated in Table 6.11.

Z_p	8	11	14	16	17	20	22	23
O_p	6 500	1 700	7 600	2 400	7 100	7 600	6 800	5 300
Z_p	24	26	27	29	30	32	33	36
O_p	7 700	7 600	6 800	4 300	7 400	5 300	7 800	7 900

TABLE 6.11: *Scheme water supply schedule when solving the single-objective model (5.2)–(5.9) in the context of the small hypothetical scenario in §6.1 implementing the parameter values proposed in §6.3.*

The irrigation schedule as well as four key performance measures for the thirty randomly generated solutions are illustrated in Table 6.12, and named according to a solution number. The total cost of scheme water supply is omitted in Table 6.12 since the total profit obtained already considers the cost of scheme water supply. It is important to note that a different convention is used to illustrate the irrigation schedule in Table 6.12. The first value in the $\left(\frac{w}{ET}\right)_{jt}$ column refers to the first crop growth stage whereas the second value refers to the second crop growth stage and so forth. The corresponding scheme water supply schedule for each solution in Table 6.11 is illustrated in Table 6.13. It is also important to note that a different convention is used to illustrate the scheme water supply schedule, and entails that each entry in the table is illustrated according to the O_{pZ_p} convention.

Solution	$\left(\frac{w}{ET}\right)_{jt}$	Profit	$V_{P_{end}}$	E_p
1	1.0, 0.59, 0.68, 0.61	R 286 322.40	130 557.0 m^3	30 911.03 m^3
2	1.0, 0.67, 0.52, 0.75	R 26 958.10	132 477.7 m^3	31 486.34 m^3
3	1.0, 0.64, 0.52, 0.65	R 13 708.88	131 315.9 m^3	32 000.11 m^3
4	1.0, 0.51, 0.55, 0.51	R 16 860.15	133 945.0 m^3	32 154.95 m^3
5	1.0, 0.64, 0.50, 0.80	R 1 861.48	130 536.8 m^3	31 527.16 m^3
6	1.0, 0.58, 0.53, 0.74	R 45 314.37	130 226.4 m^3	31 443.63 m^3
7	1.0, 0.74, 0.53, 0.63	R 18 999.09	133 576.7 m^3	31 577.27 m^3
8	1.0, 0.50, 0.69, 0.79	R 386 830.30	125 661.4 m^3	29 686.56 m^3
9	1.0, 0.52, 0.60, 0.65	R 146 489.20	129 456.3 m^3	31 405.65 m^3
10	1.0, 0.77, 0.53, 0.51	R 12 200.75	128 609.2 m^3	32 686.81 m^3
11	1.0, 0.51, 0.51, 0.73	R 230.22	130 381.0 m^3	31 582.97 m^3
12	1.0, 0.71, 0.51, 0.69	R 18 022.9	128 352.6 m^3	31 903.35 m^3
13	1.0, 0.60, 0.53, 0.51	R 680.64	128 335.9 m^3	32 724.11 m^3
14	1.0, 0.78, 0.51, 0.65	R 6 817.79	127 523.2 m^3	32 434.78 m^3
15	1.0, 0.54, 0.52, 0.66	R 11 833.50	131 962.0 m^3	31 587.97 m^3
16	1.0, 0.59, 0.53, 0.57	R 2 050.46	135 120.9 m^3	32 127.11 m^3
17	1.0, 0.53, 0.52, 0.68	R 17 782.78	128 994.6 m^3	31 905.39 m^3
18	1.0, 0.72, 0.52, 0.75	R 57 390.66	126 072.4 m^3	31 221.64 m^3
19	1.0, 0.63, 0.53, 0.55	R 3 532.49	135 333.5 m^3	32 514.55 m^3
20	1.0, 0.64, 0.54, 0.53	R 8 540.49	134 371.8 m^3	32 612.24 m^3
21	1.0, 0.84, 0.54, 0.50	R 827.13	135 155.0 m^3	32 161.02 m^3
22	1.0, 0.50, 0.58, 0.79	R 131 192.60	134 674.7 m^3	31 107.31 m^3
23	1.0, 0.79, 0.51, 0.70	R 11 617.54	130 231.0 m^3	31 269.00 m^3
24	1.0, 0.59, 0.62, 0.74	R 216 951.2	133 058.9 m^3	30 771.08 m^3
25	1.0, 0.65, 0.52, 0.72	R 15 000.91	134 426.4 m^3	31 775.57 m^3
26	1.0, 0.77, 0.52, 0.65	R 7 308.01	135 029.7 m^3	31 670.30 m^3
27	1.0, 0.61, 0.51, 0.62	R 715.3302	125 927.5 m^3	31 784.46 m^3
28	1.0, 0.52, 0.52, 0.85	R 17 050.29	130 313.0 m^3	31 883.02 m^3
29	1.0, 0.50, 0.60, 0.70	R 162 131.80	129 008.6 m^3	31 271.36 m^3
30	1.0, 0.52, 0.54, 0.52	R 3 010.91	134 493.6 m^3	32 006.40 m^3

TABLE 6.12: *Thirty random solutions generated as input to the single-objective model (5.2)–(5.9). Each solution includes the decision variable* $\left(\frac{w}{ET}\right)_{jt}$*, the total profit (in Rands), the end-period reservoir water capacity* $V_{P_{end}}$ *(in* m^3 *of water), and the total evaporation* E_p *(in* m^3 *of water).*

From Table 6.12, solution 8 achieved the largest profit of R 386 830.30 for an initial feasible solution, an end-period reservoir water capacity of 125 661.4 m^3, and a total evaporation of 29 686.56 m^3. When comparing the solution returned as output by the single-objective model (5.2)–(5.9) with solution 8 in Table 6.12, it is found that the total profit from the random feasible solution was improved on by 331.3%. Moreover, the end-period reservoir water capacity obtained as output by the single-objective model (5.2)–(5.9) lies closer to the specified user end-period reservoir water capacity, and achieved 125 081.78 m^3 of water. Also, less evaporation took place when compared to solution 8 as achieved by the single-objective model (5.2)–(5.9).

In conclusion, the solution proposed by the single-objective model (5.2)–(5.9) effectively allocates irrigation within crop growth stages so as to maximise profit, while simultaneously scheduling scheme water supply to such an extent that evaporation losses are minimised when compared to the thirty random generated feasible solutions in Table 6.12.

Solution	Scheme water supply schedule
1	$6900_{11}, 4900_{15}, 6900_{17}, 5300_{18}, 2600_{19}, 4900_{20}, 3400_{22}, 3700_{23}, 4300_{26}, 7100_{28}, 8000_{29}, 1000_{32}, 4300_{33}, 3000_{34}$
2	$2900_{11}, 1900_{12}, 5700_{14}, 4000_{15}, 4100_{18}, 4600_{19}, 3300_{22}, 1600_{24}, 5600_{25}, 2100_{26}, 6800_{27}, 3400_{28}, 2900_{32}, 6600_{33}, 3200_{34}, 7100_{35}$
3	$5600_{11}, 5700_{12}, 2300_{15}, 3700_{20}, 6100_{21}, 5800_{22}, 8000_{23}, 1800_{25}, 4500_{29}, 3700_{30}, 2900_{33}, 6600_{35}$
4	$1800_{12}, 5600_{14}, 2900_{15}, 4300_{16}, 2600_{19}, 2200_{20}, 7800_{24}, 4400_{26}, 5800_{27}, 2900_{29}, 2400_{32}, 4700_{35}$
5	$2900_{11}, 4100_{12}, 5300_{14}, 2300_{16}, 8000_{20}, 2100_{22}, 4000_{23}, 5000_{24}, 5000_{26}, 2600_{27}, 4100_{28}, 5800_{29}, 4300_{31}, 2500_{33}, 4300_{35}$
6	$2100_{11}, 7000_{12}, 5200_{15}, 4000_{19}, 7400_{22}, 6100_{23}, 6800_{24}, 2400_{29}, 6300_{30}, 3100_{31}, 4400_{32}, 3800_{34}$
7	$1600_{13}, 3700_{14}, 6700_{16}, 7700_{22}, 2300_{23}, 6800_{24}, 5700_{27}, 3400_{28}, 6200_{30}, 7400_{31}, 2600_{32}, 2600_{33}, 5000_{34}, 5200_{35}$
8	$7700_{14}, 3500_{19}, 5800_{20}, 7200_{21}, 3200_{22}, 2800_{26}, 2400_{27}, 6200_{28}, 6700_{31}, 6800_{32}, 5400_{34}, 4300_{35}$
9	$1400_{11}, 7500_{12}, 7800_{14}, 3900_{17}, 2000_{18}, 2700_{19}, 8000_{20}, 4900_{22}, 2500_{24}, 2000_{25}, 2900_{28}, 3300_{31}, 7700_{33}$
10	$3600_{11}, 3100_{12}, 4600_{13}, 5300_{14}, 4600_{15}, 2900_{16}, 2000_{18}, 5200_{19}, 4200_{23}, 4400_{25}, 3300_{28}, 6400_{29}, 4300_{33}, 3900_{34}$
11	$4300_{11}, 5300_{12}, 4200_{14}, 4700_{21}, 1800_{22}, 4700_{23}, 4200_{25}, 7200_{26}, 4800_{29}, 1800_{30}, 3600_{32}, 3100_{34}, 2300_{35}$
12	$1500_{10}, 6000_{12}, 6600_{15}, 4300_{16}, 5500_{22}, 4000_{23}, 1900_{24}, 7300_{25}, 4900_{26}, 6700_{30}, 4300_{31}, 6100_{36}$
13	$7300_{11}, 4100_{12}, 5500_{13}, 3500_{15}, 3700_{17}, 1600_{18}, 3300_{19}, 4700_{20}, 2200_{23}, 2900_{26}, 5500_{27}, 4000_{32}$
14	$5400_{11}, 6400_{12}, 2500_{13}, 8000_{14}, 1400_{15}, 7700_{16}, 7500_{26}, 5100_{27}, 8000_{28}, 3000_{29}, 4000_{32}, 4200_{36}$
15	$1300_{11}, 5300_{17}, 4500_{18}, 5300_{19}, 4300_{20}, 2900_{24}, 6800_{25}, 5500_{27}, 2300_{29}, 6100_{32}, 6100_{33}$
16	$3900_{15}, 4500_{17}, 4700_{18}, 2300_{20}, 5400_{21}, 4100_{22}, 6300_{23}, 3600_{24}, 1200_{27}, 4400_{28}, 3000_{31}, 8000_{32}, 1900_{33}$
17	$2900_{13}, 7800_{14}, 5000_{15}, 2000_{16}, 2200_{17}, 2700_{19}, 3600_{22}, 7600_{24}, 8000_{25}, 4000_{26}, 4200_{30}$
18	$1300_{10}, 3400_{11}, 2400_{17}, 4100_{19}, 7300_{20}, 6300_{22}, 4300_{25}, 7100_{26}, 7600_{28}, 1300_{30}, 8000_{31}, 3800_{33}, 5200_{36}$
19	$7600_{11}, 4400_{12}, 5100_{18}, 3700_{19}, 2700_{20}, 6900_{21}, 6400_{27}, 5700_{28}, 6500_{29}, 1700_{30}, 3200_{34}$
20	$2400_{10}, 5700_{11}, 6700_{14}, 1900_{16}, 7600_{18}, 3800_{19}, 6700_{20}, 4500_{23}, 4700_{29}, 3400_{33}, 2800_{34}, 6400_{36}$
21	$1200_{11}, 6700_{16}, 3000_{19}, 3600_{20}, 7900_{22}, 8000_{23}, 4300_{25}, 6500_{26}, 6200_{27}, 7100_{28}, 4800_{29}, 5000_{31}, 1800_{32}, 4300_{34}$
22	$6300_{11}, 1900_{12}, 7600_{14}, 4700_{19}, 7200_{20}, 2100_{21}, 7700_{23}, 2100_{24}, 4200_{25}, 5600_{27}, 2200_{30}, 1600_{33}, 6000_{34}, 4500_{35}$
23	$2900_{12}, 4000_{17}, 5400_{21}, 6800_{22}, 2200_{23}, 2600_{24}, 4800_{26}, 6000_{27}, 7300_{30}, 6800_{31}, 7500_{32}, 6500_{35}, 5100_{36}$
24	$7500_{14}, 1500_{16}, 6200_{17}, 5000_{20}, 7200_{21}, 2000_{22}, 8000_{23}, 6200_{25}, 4900_{26}, 5500_{29}, 7400_{30}, 2600_{34}, 4500_{35}$
25	$6600_{13}, 4900_{15}, 2900_{16}, 6200_{20}, 2300_{21}, 6300_{22}, 3400_{23}, 7400_{24}, 5600_{27}, 7800_{30}, 7900_{32}, 2900_{34}$
26	$2600_{14}, 6600_{18}, 7400_{19}, 6100_{21}, 8000_{22}, 4700_{28}, 4000_{29}, 6900_{30}, 4100_{31}, 4900_{32}, 2400_{33}, 3600_{35}, 7100_{36}$
27	$5000_{10}, 2200_{15}, 2900_{17}, 2000_{18}, 5900_{19}, 1600_{20}, 2200_{22}, 2900_{24}, 5400_{25}, 4900_{26}, 4100_{27}, 1700_{31}, 2900_{32}, 5500_{34}$
28	$1900_{10}, 6100_{12}, 2000_{13}, 3400_{15}, 4200_{18}, 5600_{20}, 6700_{23}, 2900_{25}, 4300_{28}, 5800_{29}, 3000_{32}, 2500_{33}, 1000_{34}$
29	$6600_{11}, 7900_{14}, 4200_{15}, 1800_{18}, 6900_{19}, 4400_{22}, 5900_{23}, 5500_{24}, 3700_{25}, 4300_{29}, 5400_{32}$
30	$1900_{10}, 3300_{13}, 2400_{17}, 2900_{18}, 5600_{20}, 2200_{21}, 4600_{22}, 2600_{23}, 5300_{25}, 3700_{29}, 3000_{32}, 5200_{33}, 5100_{36}$

TABLE 6.13: *Thirty random generated solutions to the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18). The scheme water supply schedule is shown for each scenario in Table 6.12 where the convention $O_{p_{Z_p}}$ was used.*

	A	B	C	D	E	F	G	H	I
p	Time period	1	2	3	4	5	6	7	8
t	Growth Stage	3	3	3	3	4	4	4	0
W_{jt}	CWR	9.9	11.3	10	8.5	7.5	6	4.2	0
V_p	Int vol	230000	214179.518	196285.176	180306.27	166534.518	153934.227	143612.826	136029.738
	Outsource $B_p + R_p + O_p$	0	0	0	0	0	0	0	0
x_p	Demand	14652	16724	14800	12580	11400	9120	6384	0
$V_p(E_p)$	Irr (abs)	215348	197455.518	181485.176	167726.27	155134.518	144814.227	137228.826	136029.738
A_{V_p}	Area	24297.3555	22929.956	21383.3082	20002.2158	18811.8926	17722.8214	16830.7197	16175.2965
$A_{V_{p+1}}$	Area	23030.9505	21484.4634	20104.1113	18914.8984	17826.5652	16934.5595	16278.9364	16175.2965
E_p	Evap	1168.48248	1170.34137	1178.90608	1191.75238	1200.29028	1201.40112	1199.08812	1207.60287
V_{p+1}	End Vol	214179.518	196285.176	180306.27	166534.518	153934.227	143612.826	136029.738	134822.135
	Time period	9	10	11	12	13	14	15	16
	Growth Stage	0	0	0	0	0	0	0	0
	CWR	0	0	0	0	0	0	0	0
	Int vol	134822.135	133602.231	132386.615	136175.039	140764.727	144870.697	143724.858	145345.65
	Outsource	0	0	5000	5800	5300	0	2700	0
	Demand	0	0	0	0	0	0	0	0
	Irr (abs)	134822.135	133602.231	137386.615	141975.039	146064.727	144870.697	146424.858	145345.65
	Area	16070.9207	15965.4817	15860.4133	16187.8552	16584.5521	16939.4403	16840.4028	16980.4916
	Area	16070.9207	15965.4817	16292.5744	16689.1621	17042.6429	16939.4403	17073.7699	16980.4916
	Evap	1219.90468	1215.61573	1211.57591	1210.31231	1194.02961	1145.8395	1079.20732	1004.6522
	End Vol	133602.231	132386.615	136175.039	140764.727	144870.697	143724.858	145345.65	144340.998
	Time period	17	18	19	20	21	22	23	24
	Growth Stage	0	0	0	0	0	0	0	0
	CWR	0	0	0	0	0	0	0	0
	Int vol	144340.998	143422.555	150375.14	155888.918	162866.35	162216.377	167236.034	166716.488
	Outsource	0	7800	6300	7700	0	5600	0	3700
	Demand	0	0	0	0	0	0	0	0
	Irr (abs)	144340.998	151222.555	156675.14	163588.918	162866.35	167816.377	167236.034	170416.488
	Area	16893.6572	16814.2742	17415.2015	17891.7697	18494.8447	18438.666	18872.5261	18827.6206
	Area	16893.6572	17488.4455	17959.7246	18557.2979	18494.8447	18922.6865	18872.5261	19147.4199
	Evap	918.442924	847.415368	786.221386	722.56794	649.973728	580.343122	519.545681	467.542973
	End Vol	143422.555	150375.14	155888.918	162866.35	162216.377	167236.034	166716.488	169948.945
	Time period	25	26	27	28	29	30	31	32
	Growth Stage	0	0	0	0	0	0	1	1
	CWR	0	0	0	0	0	0	3.1	9
	Int vol	169948.945	174915.342	174505.818	174110.511	179309.634	185782.403	185319.735	186214.766
	Outsource	5400	0	0	5600	6900	0	7600	4200
	Demand	0	0	0	0	0	0	6200	18000
	Irr (abs)	175348.945	174915.342	174505.818	179710.511	186209.634	185782.403	186719.735	172414.766
	Area	19107.0091	19536.2658	19500.8698	19466.7025	19916.0743	20475.5302	20435.5407	20512.9002
	Area	19573.7431	19536.2658	19500.8698	19950.723	20512.4566	20475.5302	20556.5458	19320.1355
	Evap	433.603208	409.523275	395.30778	400.876995	427.23026	462.668585	504.968822	540.69715
	End Vol	174915.342	174505.818	174110.511	179309.634	185782.403	185319.735	186214.766	171874.069
	Time period	33	34	35	36				
	Growth Stage	2	2	2	2				
	CWR	8.4	7.3	6.3	7.7				
	Int vol	171874.069	163657.872	155200.032	143737.778				
	Outsource	6800	4700	0	2000				
	Demand	14448	12556	10836	13244				
	Irr (abs)	164226.069	155801.872	144364.032	132493.778				
	Area	19273.4019	18563.2577	17832.2277	16841.5196				
	Area	18612.3682	17884.2461	16895.6481	15869.6756				
	Evap	568.196985	601.839758	626.254329	636.796164				
	End Vol	163657.872	155200.032	143737.778	131856.982				

FIGURE 6.6: *The estimated total evaporation and the end-period reservoir water capacity executed using Excel as a validation. The irrigation schedule in Table 6.8 and the scheme water supply schedule in Table 6.9 were manually inserted into the Excel model of which the aforementioned results were then calculated.*

6.4.3 An expert validation

The final method of validation involved consulting a specialist in the field of crop irrigation and farming. Ruan Gerber [55] was asked to fulfil the role as expert in the field based on his qualifications and responsibilities at his work place. Gerber works at Breërivier Irrigation located in Robertson and has a BEng (Civil) Engineering degree which he obtained in 2016 from Stellenbosch University. Gerber also obtained his MSAII (Designer) course in 2017 and is also a SABI irrigation systems designer. Currently, Gerber is the only approved irrigation designer at Breërivier Irrigation and is responsible for all the irrigation systems developed at Breërivier Irrigation.

Gerber's work description demands having a wide spectrum of knowledge with respect to the growth of crops, crop water requirements, waterholding capacity of soils, topography of the property, and so forth. Primarily, Gerber develops and installs irrigation systems in the Robertson region while accepting responsibility for all irrigation systems developed at Breërivier Irrigation. One of the recent projects Gerber was involved in entailed the development and installation of an irrigation system for citrus for Habata Boerdery (Pty) Ltd located 20km outside Robertson. Habata currently grows citrus on a 1000 hectares and vegetables on 700 hectares in the Eastern and Western Cape of South Africa.

During an interview with Gerber, the DSS was thoroughly discussed along with the working of the DSS based on the two mathematical models and the model framework. Given that a similar farming scenario is solved to the hypothetical scenario described in §6.1, Gerber concluded that the results obtained by the DSS are realistic in nature. Specific emphasis was placed upon determining the crop water requirements, and it was found that Gerber uses SAPWAT, with which CropWat is affiliated to, to determine the crop water requirements for crops. This is done by considering the crop coefficients. In this thesis, a similar approach was taken when estimating the crop water requirements for the crops, as in the small hypothetical scenario in §2.4.6.

Furthermore, Gerber's opinion was asked with regards to the irrigation of crops when water supply is limited. He concluded that irrigation should be reduced to the least profitable crop with respect to the yield response factors associated with the growth stages of that crop. When irrigating a crop during a less sensitive growth stage, the crop may still produce yield while remaining irrigation is then supplied to more profitable crops. Gerber was not asked to solve the hypothetical scenario described in §2.4.6 as it would be too complex and time consuming to solve the problem by hand. Considering the aforementioned suggestion by Gerber, it is clear that the solution proposed by the single-objective model (5.2)–(5.9) resulted in reducing irrigation during the crop growth stage with the lowest yield response factor — the solution proposed by the mathematical model corresponds with Gerber's suggestion.

One advantage of incorporating the single-objective model (5.2)–(5.9) and/or the bi-objective crop model (5.10)–(5.18) is that a more accurate solution is proposed as a solution for irrigating crops under a limited water environment. The optimal solution considers multiple feasible solutions where irrigation is reduced in multiple crop growth stages, of which the affect thereof is noted in the objective function. The objective function also considers the cost of acquiring additional water resources as part of the total profit as well as the total evaporation when scheduling additional water resources using scheme water supply. Higher evaporation results in buying more additional water resources which then affects the total profit.

Moreover, Gerber also concluded that his planning is based on extreme weather conditions. This include developing crop irrigation systems and computing crop water requirements for the proposed irrigation systems while assuming that no rainfall will occur. Finally, the single-objective model (5.2)–(5.9) and bi-objective model (5.10)–(5.18), as well as the working of the

DSS, which will be thoroughly discussed later on, and the model framework was comprehensively presented to Gerber. Given Gerber's qualifications and work responsibilities, he was able to provide critical insights on the working of the DSS. In conclusion, Gerber expressed that he found no obvious errors with regards to the working of the DSS, the initial results obtained, estimating the key performance measures, the shape characteristic used as a hypothetical reservoir and the evaporation data.

6.5 Chapter summary

This chapter was devoted to an illustration of the working of the single-objective model (5.2)–(5.9) and bi-objective model (5.10)–(5.18) by solving them in the context of a small hypothetical scenario. This chapter opened in §6.1 with a brief description of the small hypothetical farming scenario in which the single-objective model (5.2)–(5.9) was solved. This scenario incorporated evaporation data as well as a shape characteristic of a reservoir in order to model the variation of reservoir volume as water is supplied and irrigated from the reservoir.

Next, in §6.2, the model implementations were thoroughly discussed with respect to the solution encoding, the way in which an initial feasible solution and neighbouring solutions were generated, the cooling and reheating schedules that were implemented, and finally, the way in which the initial temperatures were estimated. Next, in §6.3, a parameter evaluation was conducted on five algorithmic parameters of which proposals were given for best model results. Moreover, in §6.4, the two mathematical models (5.2)–(5.9) and (5.10)–(5.18) were validated in three ways, namely a face validation, a random benchmark validation and an expert validation in the field of crop irrigation and farming.

From the expert validation, it was found that irrigation systems planning commences on the assumption that no rainfall may occur, which was an essential assumption on which the mathematical models (5.2)–(5.9) and (5.10)–(5.18) were formulated on. It was also found that SAPWAT is used during the process of developing irrigation systems, and similar in this thesis, CropWat 8.0 was used to determine parameters that play an important role in estimating the final crop yield. Finally, the chapter was closed in §6.5 with a brief review of the chapter contents.

CHAPTER 7

Case study

Contents

In this chapter, the indispensable quality of the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18) proposed in Chapter 5 is illustrated by solving the mathematical models in the context of a realistic farm scenario. The first section in this chapter introduces a case study as part of a realistic farm scenario. This is followed in §7.2 by solving the single-objective model (5.2)–(5.9) and bi-objective model (5.10)–(5.18) in the context of the hypothetical scenario introduced in §7.1 by incorporating the parameter values proposed in §6.3. In this section, the results obtained from solving the aforementioned models are thoroughly discussed. Finally, the chapter closes in §7.3 with a brief summary of the chapter contents.

7.1 A realistic farm scenario

The realistic farming scenario developed here entails proposing an irrigation schedule along with a scheme water supply schedule that is provided as decision support to farmers in order to irrigate their crops when water supply is limited. For this scenario, three different crops are considered, namely wheat, maize and potatoes which are grown on a farm in the South Eastern Swartland district area.

As mentioned in §6.1, maize is considered as the most important grain crop in South Africa of which are grown in a number of locations in the Western Cape as a result of high silage prices globally [39, 45]. This crop, however, can only be grown if sufficient water resources are available for irrigation due to high water demand and is usually irrigated using a center pivot irrigation[1]. Wheat, on the other hand, is considered as an important cereal crop in South Africa and ranks second after maize in terms of area planted and production quantities delivered when harvested.

[1]Center pivot irrigation is a method of crop irrigation where crops are irrigated using sprinklers by equipment that rotates around a pivot [68].

Currently, the Free State is the highest producer of wheat followed by the Western Cape and then the Northern Cape. In total, the demand for wheat in South Africa is approximately 2.7 million tons annually of which only 2 million tons are produced locally [30]. This calls for importing wheat from Argentina, the United States of America, the United Kingdom, and the Ukraine to name a few.

Wheat is typically grown in the south western parts of the Western Cape, and resulted in approximate 697 000 tons of wheat produced for the 2015 production season [30]. The average ton per hectare for wheat grown in the Western Cape for the production season 2014–2015 was approximately 2.90 ton per hectare. This is significantly lower than the average ton per hectare produced in the Northern Cape which was approximately 7.50 ton per hectare. Wheat that is grown in the Western Cape, however, is typically produced on dry land implying that the crop is grown without applying irrigation to it. This may explain the significant lower yields obtained for wheat in the Western Cape when compared to the yields obtained in the Northern Cape.

Moreover, Potatoes are grown in sixteen different regions across South Africa and, therefore, ensures a constant supply of potatoes throughout the year. Over a number of years, potatoes has been grown on dry land, however, recent studies show that 45 000 hectares are under irrigation whereas approximately 7 500 hectares are grown on dry land [99]. Furthermore, average yields for potatoes in South Africa has more than doubled when compared to the production in year 1990 to approximately 40 ton per hectare. To place this into perspective, in the Africa content, South Africa is responsible for 3.5% of the area under production whereas South Africa contributes approximately 11% of the total potato production, which shows the effective practices applied in South Africa when compared to other countries for growing potatoes.

Considering the three crops, the farmer's primary income resonates from producing wheat, however, also plants yellow maize to produce animal feed to the local market. Furthermore, the farmer also grows potatoes on a number of hectares in order to improve cash flow to reduce the risk if a bad wheat harvest is experienced. Wheat is typically dry land produced when grown in the Western Cape, however, in this case it is taken that wheat is irrigated since it is assumed that no rainfall will occur. Furthermore, the farmer has one irrigation reservoir which is used to irrigate the number of crops grown on the farm while storing additional water from scheme water supply. The reservoir has a water capacity of 550 000 m^3 with a similar shape characteristic as in Figure 6.1. The farmer also indicated that the current reservoir water capacity is at 86%. The shape characteristic considered in this farming scenario is illustrated in Table 7.1.

Assuming further, that the farmer has been granted water rights to pump additional water resources from the river, and is entitled to 50% of the total water requirements of the crops grown on the farm. Moreover, the time continuum taken into account is one hydrological year. This resulted in a total scheme water supply for one hydrological year of 212 000 m^3, of which a maximum of 20 000 m^3 of water per time period may be acquired and a minimum of 2 000 m^3 of water per time period may be acquired. In Table 7.2, a summary of the reservoir related parameters as described above is illustrated.

The production costs associated with growing wheat and maize were gathered from the Grain SA [58] website. The production costs associated with growing potatoes was gathered from Troskie [131] which evaluated the impact of electricity price increase on the potato industry as well as the Potato South Africa [100] website. Moreover, the water cost is taken as R 3.67 per m^3 in this scenario. A summary of the demand-related parameters is illustrated in Table 7.3 for this scenario where the water cost is also indicated. The parameters shown in Table 7.3 include the average yield (in ton per hectare) obtained for each respective crop type, the selling price for each crop type (in Rands), the number of hectares planted for each crop type, the variable cost for each crop type (in Rands per hectare) and also the fixed cost for each crop type (in Rands).

Reservoir volume (m^3)	Reservoir surface area (m^2)
1911.64	603.50
3905.53	3473.73
13178.20	7969.78
29729.64	12095.89
53559.98	15964.30
68204.68	17836.95
84669.10	19687.88
102953.24	21531.56
123057.05	23380.68
144980.58	25250.91
168723.84	27154.94
194286.76	29107.86
221669.41	31124.14
250871.47	33216.47
281893.87	35418.63
314735.62	37688.98
349397.10	40096.93
385878.24	42638.86
424179.11	45327.44
464299.70	48178.36
506240.02	51204.29
550000.00	54420.33

TABLE 7.1: *The hypothetical shape characteristic of an irrigation reservoir considered in the realistic farming scenario.*

Reservoir capacity	520 000 m^3
Current water level	86%
O_{total}	212 000 m^3
O_{max}	20 000 m^3
O_{min}	2 000 m^3
$V_{P_{end}}$	30%

TABLE 7.2: *The reservoir-related parameters as taken in the realistic farming scenario.*

Crop	Average Yield	Selling price	Hectares	Variable cost	Fixed cost
Wheat	4.5 ton per ha	R 3 170	110 ha	R 7 752 per ha	R 2 745
Maize	9.0 ton per ha	R 2 360	25 ha	R 7 520 per ha	R 2 456
Potato	42 ton per ha	R 421	45 ha	R 9 845 per ha	R 4 326

TABLE 7.3: *Farming-related parameters as taken in the realistic farming scenario.*

Moreover, the demand-related parameters were gathered from CropWat 8.0 for wheat, maize and potatoes according to the nearest weather station. These parameters are illustrated in Table 7.4 where the month of growth, the crop growth stage for that month, the crop water requirements during each crop growth stage ET_c, and the yield response factor K_{jt} are shown. Note that the time chart for planting potatoes according to [112] stretches from January till September, however, in this case potatoes are harvested a month later. This is done to minimise the number of periods overlapping when growing wheat and potatoes since irrigation requirements during these time periods may be high. Furthermore, harvesting potatoes a month later provides more

financial security in terms of supplying food outside the window of potato growth since the demand is high given that supply is low.

Finally, the historical A-pan evaporation rates for the Voëlvlei dam were gathered from Weather South Africa [142] which is located in the Eastern Swartland district area. This data was then used to estimate the total evaporation from the reservoir water surface area. A polynomial function is fitted to the evaporation rates gather for the Voëlvlei dam, as done in §6.1, of which the resulting 7^{th} degree polynomial function for the historical daily average evaporation rate is illustrated in Figure 7.1. The degree of fitting was chosen as such according to the R^2 values obtained from the fit, as done in §6.1.

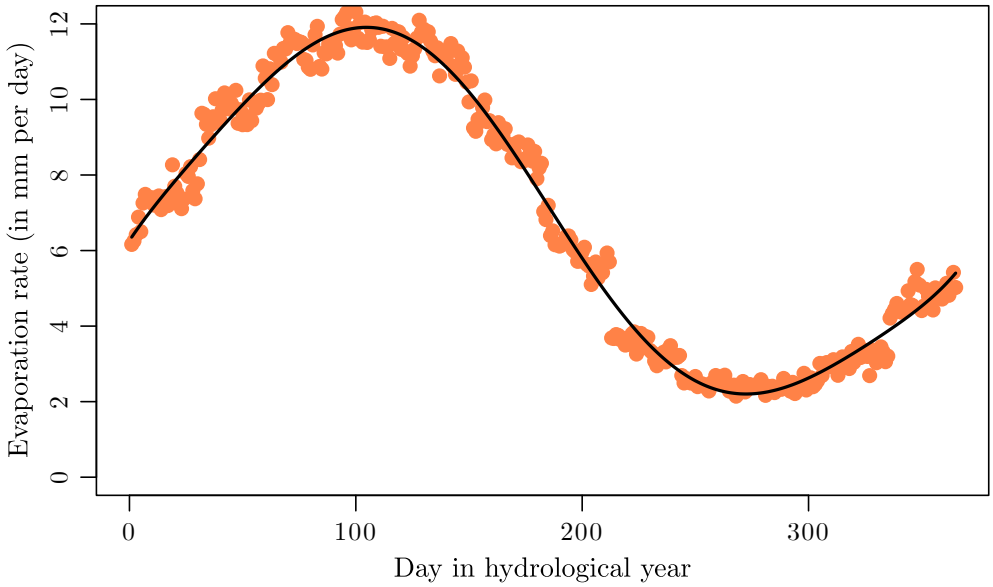


FIGURE 7.1: *A 7^{th} degree polynomial function fitted to the historical daily average evaporation rate experienced at the Voëlvlei dam.*

7.2 Numerical results

In this section, the results obtained from solving the single-objective model (5.2)–(5.9) and the bi-objective model (5.10)–(5.18) in the context of the hypothetical farm scenario described in the previous section are comprehensively discussed. Moreover, a reflection is also provided on the results obtained from the aforementioned mathematical models.

7.2.1 Results for the single-objective model (5.2)–(5.9)

First, the single-objective model (5.2)–(5.9) was solved by employing the SA algorithm, as described in §4.3.1, using the proposed parameter values, as described in §6.3, as algorithmic parameters. The resulting key performance measures when solving this model are illustrated in Table 7.5 as obtained from the model.

In Table 7.5, a total profit of R 622 632.83, an end-period reservoir water capacity of 165 001.83 m^3, a total scheme water cost of R 767 397.00 (at R 3.67 per m^3 of water) was achieved while a total evaporation of 95 852.97 m^3 took place over the scheduling horizon. Moreover, it seen that the end-period reservoir water capacity of 30%, as specified by the farmer, and a total

Wheat				Maize				Potato			
Month	Growth stage	ET_c	K_{jt}	Month	Growth stage	ET_c	K_{jt}	Month	Growth stage	ET_c	K_{jt}
Apr	Initial	6.5mm	0.40	Nov	Initial	8.4mm	0.40	Jun	Initial	3.7mm	0.45
Apr	Initial	11.1mm	0.40	Nov	Initial	14.7mm	0.40	Jun	Initial	6.2mm	0.45
May	Initial	11.2mm	0.40	Dec	Development	18.3mm	0.40	Jul	Development	6.2mm	0.80
May	Development	11.4mm	0.60	Dec	Development	32.7mm	0.40	Jul	Development	8.0mm	0.80
May	Development	12.3mm	0.60	Dec	Development	53.5mm	0.40	Jul	Development	13.4mm	0.80
Jun	Development	13.8mm	0.60	Jan	Middle	64.4mm	1.30	Aug	Middle	17.1mm	0.80
Jun	Middle	14.9mm	0.80	Jan	Middle	69.5mm	1.30	Aug	Middle	19.8mm	0.80
Jun	Middle	13.8mm	0.80	Jan	Middle	74.7mm	1.30	Aug	Middle	25.7mm	0.80
Jul	Middle	12.4mm	0.80	Feb	Middle	66.3mm	1.30	Sep	Middle	27.0mm	0.80
Jul	Middle	11.5mm	0.80	Feb	Late	63.7mm	0.50	Sep	Middle	30.5mm	0.80
Jul	Late	11.1mm	0.40	Feb	Late	39.0mm	0.50	Sep	Late	32.9mm	0.30
Aug	Late	11.0mm	0.40	Mar	Late	32.8mm	0.50	Oct	Late	32.3mm	0.30
Aug	Late	10.7mm	0.40	Mar	Late	16.6mm	0.50	Oct	Late	30.8mm	0.30
Aug	Late	1.7mm	0.40					Oct	Late	6.2mm	0.30

TABLE 7.4: *The demand-related parameters for crops wheat, maize and potatoes. This includes the month of grow, the crop growth stage during each month, and the crop water requirements during each month for each respective crop.*

Total profit	R 611 418.92
End period reservoir volume	165 001.83 m^3
Total scheme water cost	R 767 397.00
Total evaporation	95 852.97 m^3

TABLE 7.5: *The maximum profit, the end-period reservoir water capacity, the total scheme water cost and the total evaporation as part of the four key performance measures obtained from solving the single-objective model (5.2)–(5.9) in the context of the hypothetical farm scenario.*

evaporation of 17.43% of the reservoir water capacity has taken place in the hydrological year. In Table 7.6, the values for the decision variable $\left(\frac{w}{ET}\right)_{jt}$ (which denotes the ratio of the amount of water supplied over the amount of water required by crop j during growth stage t) as obtained by the single-objective model (5.2)–(5.9) are illustrated.

Crop	Crop growth stage t			
j	1	2	3	4
Wheat	100%	100%	100%	100%
Maize	100%	100%	98%	98%
Potato	100%	100%	100%	98%

TABLE 7.6: *The values obtained for the decision variable $\left(\frac{w}{ET}\right)_{jt}$ when solving the single-objective model (5.2)–(5.9) in the context of the case study. Each value shows the irrigation (in %) in crop growth stages for specific crops.*

The irrigation schedule propose that 100% of the crop water requirements of wheat should be irrigated in all its growth stages, 100% of the crop water requirements of maize should be irrigated in growth stages 1 and 2 while 98% of the crop water requirements of maize should be irrigated in growth stages 3 and 4. Finally, for potatoes 100% of the crop water requirements should be irrigated in growth stages 1–3 while 98% of the crop water requirements of potatoes should be irrigated in growth stage 4. Moreover, the resulting irrigation schedule in Table 7.6, therefore, suggests that irrigation should be lowered for maize in the third and fourth crop growth stage by 2%, respectively, and that irrigation should also be lowered for potatoes in the fourth crop growth stage by 2%.

Recall from §2.4.2, that a yield response factor K_{jt} with a value larger than 1 may result in large yield reductions when experiencing a water deficit. In Table 7.6, it is interesting to see that irrigation is reduced in the third crop growth stage (*i.e.* the growth stage called "Middle" in Table 7.6) for maize since it is the growth stage with the highest yield response factor — for the first and second growth stage $K_{j1-2} = 0.40$, for the third growth stage $K_{j3} = 1.30$ and for the fourth growth stage $K_{j4} = 0.50$. Moreover, the crop water requirement ET_c during each month is 23.1 mm for the first growth stage, 104.5 mm for the second growth stage, 274.9 mm the third growth stage and 152.1 mm for the fourth growth stage. Therefore, it is noted that the third crop growth stage for maize demands the highest water compared to the water requirements during other months.

The irrigation schedule for maize suggest that water should be reduced during the third crop growth stage, which is the most sensitive growth stage but also the growth stage with the highest amount of water required. It may, therefore, be noted for this specific crop, that it is more profitable to reduce irrigation to maize in the most sensitive crop growth stage and irrigate the high amounts of unused water as a result of the high crop water requirements during this growth stage to other crops during its growth stages. The model, therefore, weighs up the reduction in actual yield for maize as a result of reduced irrigation in the third crop growth stage against

irrigating water to other crops in growth stages while considering the effect that evaporation has on the reservoir water capacity and the water availability (*i.e.* storing the unused water in the reservoir to be irrigated at a later stage). Moreover, irrigation is also reduced during the fourth crop growth stage (*i.e.* the growth stage called "Late" in Table 7.6) of maize which demands the second highest amount of water compared to the crop water requirements for the first, second and third crop growth stage. In this case, the yield response factor K_{jt} is considerably lower than the yield response factor of the fourth growth stage, and may, therefore, explain why irrigation is reduced during this growth stage. As previously mentioned, irrigation lowering during the fourth growth stage seems to be more profitable as it is irrigated to other crops — the high water requirements during this growth stage results in high volumes of water that may be irrigated to crops in other growth stages.

Moreover, the irrigation schedule for potatoes also suggest that irrigation should be reduced during the fourth crop growth stage. From Table 7.6, $K_{j1} = 0.45$ for the first growth stage, $K_{j2-3} = 0.80$ for the second and third growth stage, and $K_{j4} = 0.30$ for the fourth growth stage for potatoes. Moreover, the crop water requirements ET_c for potatoes for the first growth stage is 9.9 mm, for the second growth stage is 27.3 mm, for the third growth stage is 120.1 mm and for the fourth growth stage is 102 mm. The suggestion of reducing irrigation in the third crop growth stage, therefore, seems of good quality given that the fourth crop growth stage has the lowest K_{jt} value. Furthermore, the fourth crop growth stage demands the second highest amount of water compared to the first, second and third growth stages. It is, however, important to note that the yield response factors for the second and third growth stages are higher as well as the crop water requirement for the third crop growth stage. By proposing this irrigation schedule, the model weighs up the affect of reducing irrigation in the second crop growth stage compared to large amounts of water gained from reducing irrigation in the third and fourth growth stages of potatoes.

Finally, the irrigation schedule for wheat proposes that 100% of the crop water requirements should be irrigated to all its growth stages, and may be suggested as such due to the fact that 110 hectares of wheat are planted of this crop compared to the 25 hectares planted for maize and 45 hectares planted for potatoes. Moreover, the total water requirements for wheat is 153.4 mm, 554.6 mm for maize and 259.8 mm for potatoes. Given that 110 hectares of wheat are planted, is also makes sense to allocate full irrigation in all of its growth stages since the crop water requirements are the lowest of the three crops. In order to irrigate the three respective crops to such a degree as proposed by the irrigation schedule in Table 7.6, high volumes of scheme water supply are required. The resulting scheme water supply schedule when solving the single-objective model (5.2)–(5.9) is illustrated using a bar chart in Figure 7.2 where the amount of scheme water supply is shown on the top of each bar during each time period.

From Figure 7.2, a total of 209 100 m^3 of additional water resources should be acquired over the scheduling horizon where a total of 18 time periods are scheduled for acquiring additional water resources from scheme water supply. The scheme water supply schedule suggest that 4 500 m^3 of water should be pumped during time period 7, 8 700 m^3 of water should be pumped during time period 8, 4 000 m^3 of water should be pumped during time period 9, 14 200 m^3 of water should be pumped during time period 15, 19 600 m^3 of water should be pumped during time period 18, 10 200 m^3 of water should be pumped during time period 19, 12 400 m^3 of water should be pumped during time period 20, 17 200 m^3 of water should be pumped during time period 21, 7 600 m^3 of water should be pumped during time period 25, 18 700 m^3 of water should be pumped during time period 26, 4 800 m^3 of water should be pumped during time period 28, 16 700 m^3 of water should be pumped during time period 30, 13 000 m^3 of water should be pumped during time period 31, 2 200 m^3 of water should be pumped during time period

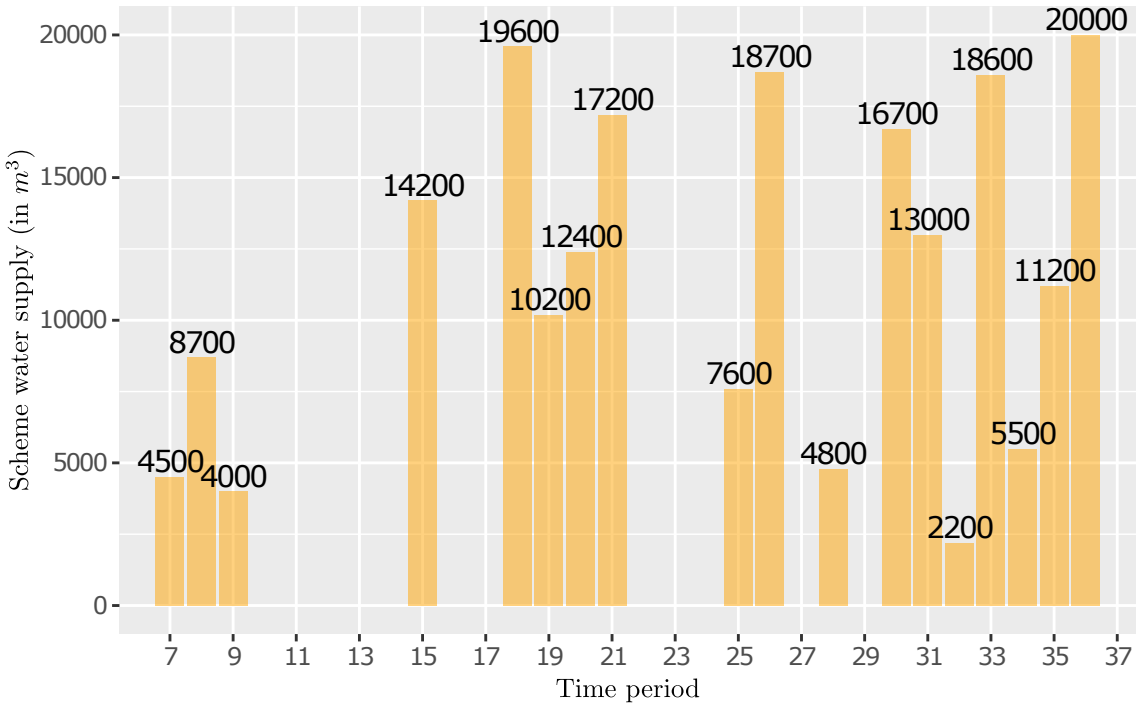


FIGURE 7.2: *The scheme water supply schedule when solving the single-objective model (5.2)–(5.9) in the context of the hypothetical farm scenario. The amount of scheme water supply that should be acquired is also indicated for each time period at the top of each bar.*

32, 18 600 m^3 of water should be pumped during time period 33, 5 500 m^3 of water should be pumped during time period 34, 11 200 m^3 of water should be pumped during time period 35, and 20 000 m^3 of water should be pumped during time period 36.

Considering the time at which scheme water supply is scheduled, it is found that 75.61% (158 100 m^3 of water) of the total scheme water supply is scheduled during time periods 19–36 (that is days 190 until 360 in Figure 7.1). This may be due to the fact that the evaporation rates are lower during these time periods since these time periods occur during the winter season. The evaporation rates are higher for time periods 0–18 as it is during the summer season given that the first time period starts on the first of October (the beginning of a hydrological year). Furthermore, it is also noted that 75.61% of scheme water supply are scheduled during the winter times which are predominant to rainfall since the Western Cape receives winter rainfall. This ensures the feasibility of the proposed solution given that water availability during these periods are higher than in the summer season, and more water are scheduled for pumping during time periods where evaporation rates are at its lowest. By doing this, less water are lost to evaporation. It may also be that water are scheduled during the aforementioned time periods due to the combined irrigation between wheat and potatoes, which overlaps by 8 time periods as seen in Table 7.4.

Taking into account the irrigation schedule in Table 7.6 and the crop water requirements in Table 7.4, an irrigation schedule may be developed which shows the exact amount of water which should be irrigated to crops during specific time periods. This schedule is computed by multiplying the percentage of crop water requirements that should be irrigated to crops in its respective growth stage with the crop water requirements for that specific growth stage as well as the number of hectares planted of the crop. For example, consider that 100% of the crop water requirements should be irrigated to wheat in the first growth stage. The total amount of water that should be irrigated during the month of April is then calculated by

$100\% \times 0.065\,m$ (that is $\frac{ET_c}{1000}$) $\times 1\,100,000\,m^2$ (that is 110 hectares multiplied by $10\,000\ m^2$ since $1\,\text{ha} = 10\,000\ m^2$). The computed irrigation schedule which shows the amount of water that should be irrigated to crops for the amount of hectares planted of that crop for a specific time period is illustrated in Figure 7.3. On the x-axis of this figure the time period is shown whereas the amount of water that should be irrigated (in m^3) is shown on the y-axis.

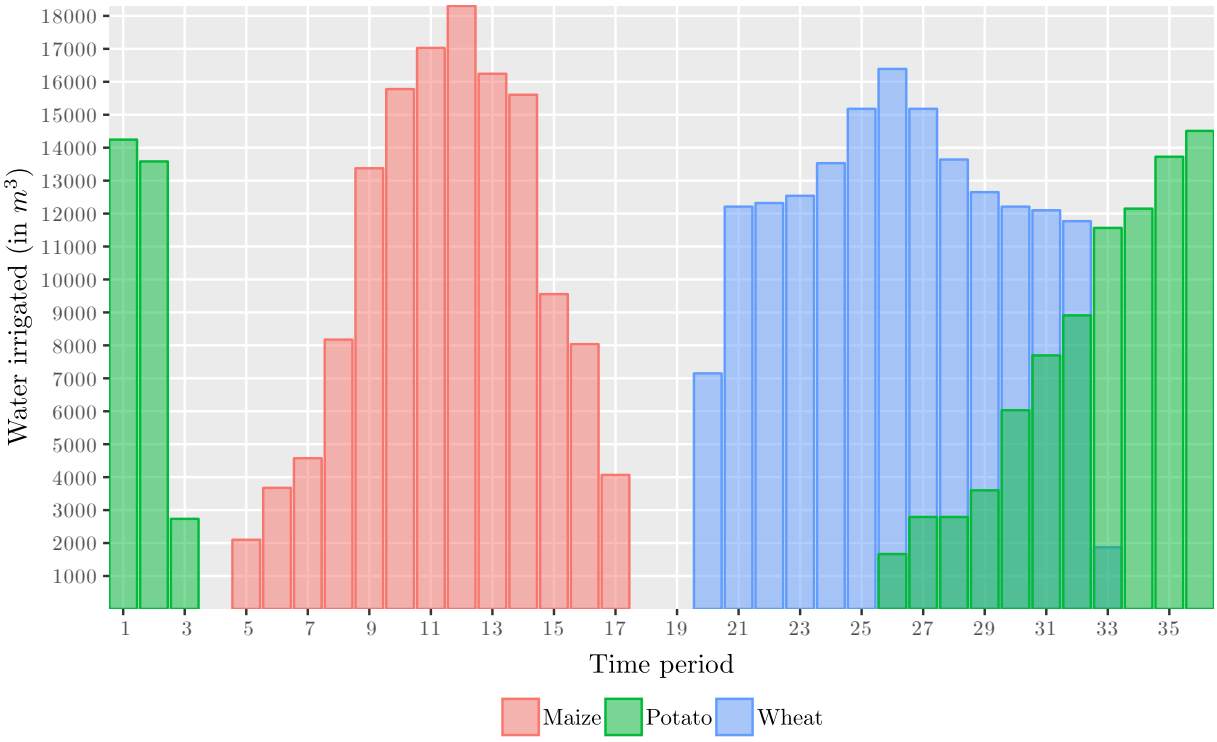


FIGURE 7.3: *The irrigation schedule obtained when solving the single-objective model (5.2)–(5.9) in the context of the hypothetical farm scenario. This schedule shows the amount of water irrigated to crops (in m^3) for all the hectares assigned to the crop during specific time periods.*

From Figure 7.3, it is found that a total of $168\,740\,m^3$ of water should be irrigated to 110 hectares of wheat, $136\,515\,m^3$ of water should be irrigated to 25 hectares of maize, and $115\,990.2\,m^3$ of water should be irrigated to 45 hectares of potatoes. The highest volume of water should, therefore, be irrigated to wheat since the largest area are planted for this crops. In this figure, it is also noted that a lot of strain is placed on the irrigation reservoir during time periods 26–33 for providing irrigation to crops given that irrigation is shared between wheat and potatoes during these time periods. Moreover, an irrigation schedule may be developed which shows the amount of water that should be irrigated to crops per hectare. This may be calculated in a similar fashion as described earlier, except that the amount of hectares assigned to each crop are now disregarded (*i.e.* the % of crop water requirements that should be irrigated to crops in its respective growth stage is multiplied with the crop water requirements for that specific growth stage). The computed irrigation schedule which shows the amount of water that should be irrigated to crops per hectare for a specific time period is illustrated in Figure 7.4.

From Figure 7.4, it is noted that maize demands the highest amount of water per hectare although irrigation is reduced during the third and fourth growth stage, that potatoes demands the second highest amount of water per hectare although irrigation is reduced during the fourth growth stage, and that wheat demands the least amount of hectare although 100% of the crop water requirements should be irrigated to wheat in all its growth stages. Moreover, considering the amount of water that should be irrigated to crops per hectare, it is noted that for maize and

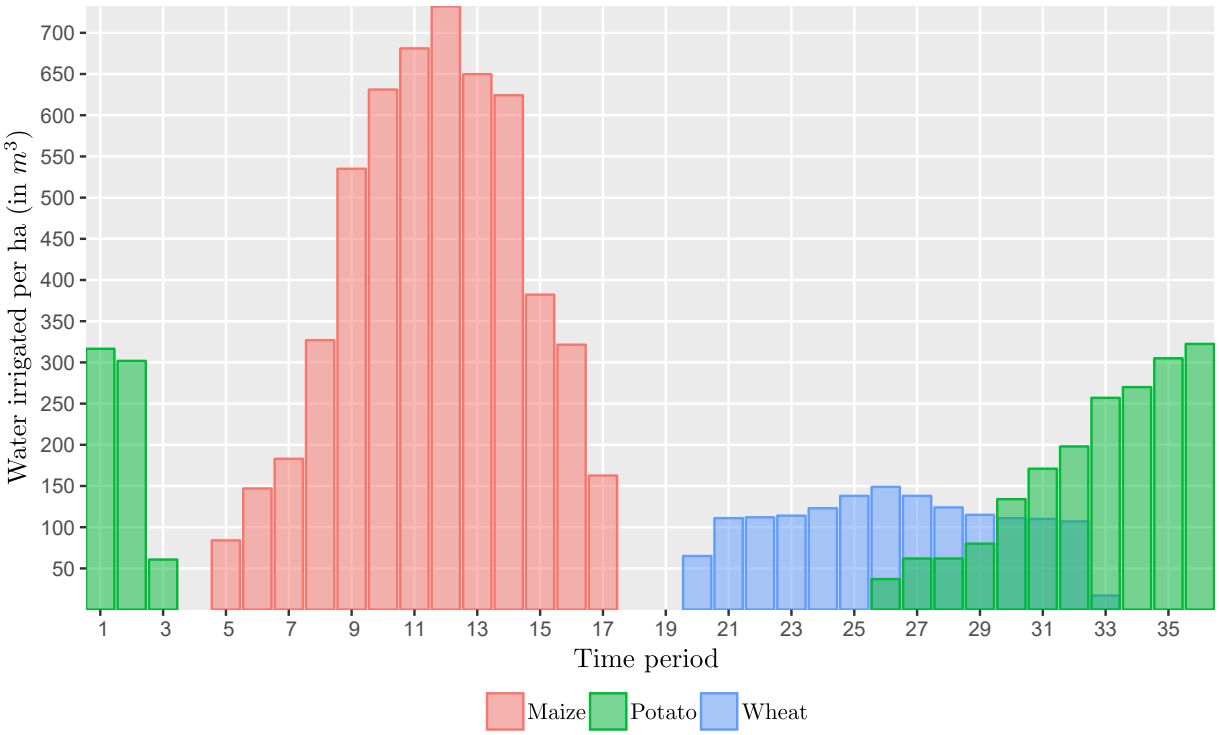


FIGURE 7.4: *The irrigation schedule obtained when solving the single-objective model (5.2)–(5.9) in the context of the hypothetical farm scenario. This schedule shows the amount of water irrigated to crops (in m^3) per hectare for each respective crop during specific time periods.*

potatoes the amount of water irrigated per hectare incrementally increases as the crop makes it's way through its growth stages while an approximate equal amount of irrigation is applied to wheat except during time periods 20 and 33. Considering that wheat are dry land grown (*i.e.* no irrigation is given to the crop) and that 100% irrigation is allocated to wheat in all its growth stages for the purpose of this hypothetical scenario, the irrigation given to wheat in Figure 7.4 is merely an indication of the expected rainfall during the time periods 20–33 when grown.

7.2.2 Results for the bi-objective model (5.10)–(5.18)

When solving the bi-objective model (5.10)–(5.18) in the context of the hypothetical scenario described in §7.1, it is expected to obtain a set of nondominated solutions which result in a Pareto front when plotted. The parameter values proposed in §6.3 are used as algorithm parameters to solve the bi-objective model (5.10)–(5.18). The following front was obtained when solving this model, and is illustrated in Figure 7.5. From this figure, it is clear that the front obtained is linear. Moreover, the nondominated solutions in this front are indicated by red dots while a geom_smooth() function in RStudio was used to apply a smoothing, as indicated by the blue line in Figure 7.5. In this figure, the total profit (in Rands) is shown on the x-axis and the end-period reservoir water capacity (in m^3 of water) is shown on the y-axis.

It was previously stressed in §6.3 that the nature of the two mathematical models described in Chapter 5 is similar to one another. It is important to note that the solution obtained from the single-objective model (5.2)–(5.9) supposedly lies on the front of optimal solutions obtained by solving the bi-objective model (5.10)–(5.18), given that the objective function of the prior model is also taken as an objective function in the latter model. The front obtained by the bi-objective model (5.10)–(5.18) is linear for the following reason. The largest portion of water required to

irrigate crops is obtained from allocating additional water resources and, therefore, ensures that small amounts of water are reduced in crop growth stages. This is clear from the irrigation schedule obtained from the single-objective optimisation model (5.2)–(5.9) in Table 7.6, and explains that only 2% of the crop water requirements are reduced to maize during growth stages 3 and 4, respectively, and only 2% of the crop water requirements are reduced to potatoes during growth stage 4, while the remaining water irrigated to the crops were obtained by allocating additional water resources. The scheme water supply and the reservoir water capacity at the beginning of the scheduling horizon accounts for 100% irrigation of the crop water requirements to wheat in all its growth stages, 100% irrigation of the crop water requirements to maize on growth stages 1 and 2 while 98% irrigation of the crop water requirements in growth stages 3 and 4, and 100% irrigation of the crop water requirements for potatoes in growth stages 1 – 3 while 98% irrigation of the crop water requirements in growth stage 4.

Essentially, the algorithm tends to allocate additional water resources rather than reducing irrigation to crops since it is more cost beneficial to do so. The non-linearity in reducing irrigation to crops is, therefore, not visible in the front illustrated in Figure 7.5 due to the small amount of water that is reduced in crop growth stages. The linear curve presented in this figure rather represents the total amount of profit obtained for multi reservoir water capacities where the large amounts of scheme water supply were allocated to irrigate crops and little amounts of water were reduced during crop growth stages.

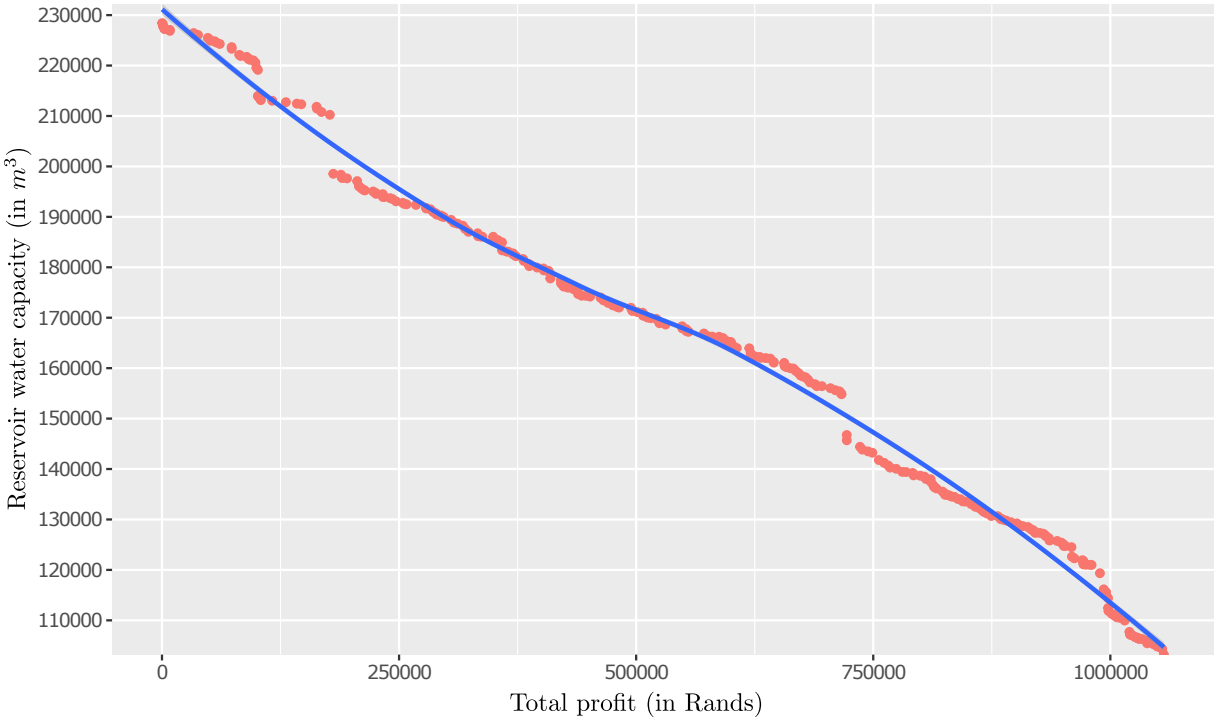


FIGURE 7.5: *The obtained linear front when solving the bi-objective model (5.10)–(5.18) in the context of the hypothetical farm scenario.*

In order to illustrate the non-linearity associated with reducing irrigation to crops as well as displaying a larger portion of the Pareto front, the hypothetical scenario in §7.1 was again solved but for less scheme water supply available. Eventually, the total scheme water supply was reduced to 160 000 m^3 of water (*i.e.* the farmer is entitled to 37.75% of the total crop water requirements compared to the 50% of crop water requirements initially) available during the scheduling horizon. When solving the case study for the new adopted maximum scheme water supply, an approximate Pareto front was obtained and is illustrated in Figure 7.6.

From this figure, it is clear that a linear curve is shown for the reservoir water capacity of approximately 166 000 m^3 to 230 000 m^3. The range of the linear curve illustrates the portion of the solutions where additional water resources are acquired to irrigate crops rather than reducing irrigation to crops. After the maximum amount of scheme water supply has been reached (that is the point where the reservoir water capacity $\approx$ 166 000 m^3), the algorithm starts to reduce irrigation to crops which results in a Pareto front of optimal solutions, as seen in Figure 7.6 for reservoir water capacities of approximate 166 000 m^3 to 90 000 m^3 of water. The range of the Pareto front illustrates the portion of solutions where irrigation water are reduced to crops while a maximum amount of scheme water supply is allocated. This is due to the fact that irrigation is reduced when the maximum scheme water supply has been reached. Moreover, a reservoir water capacity of approximately 90 000 m^3 of water is left over if the profit is to be maximised without considering the end-period reservoir water capacity. This solution lies in the bottom right hand corner in Figure 7.6 and achieved a total profit of approximately R 1 060 000. In real-life, a tailored farm scenario may result in either of the Pareto fronts in Figures 7.5 and 7.6.

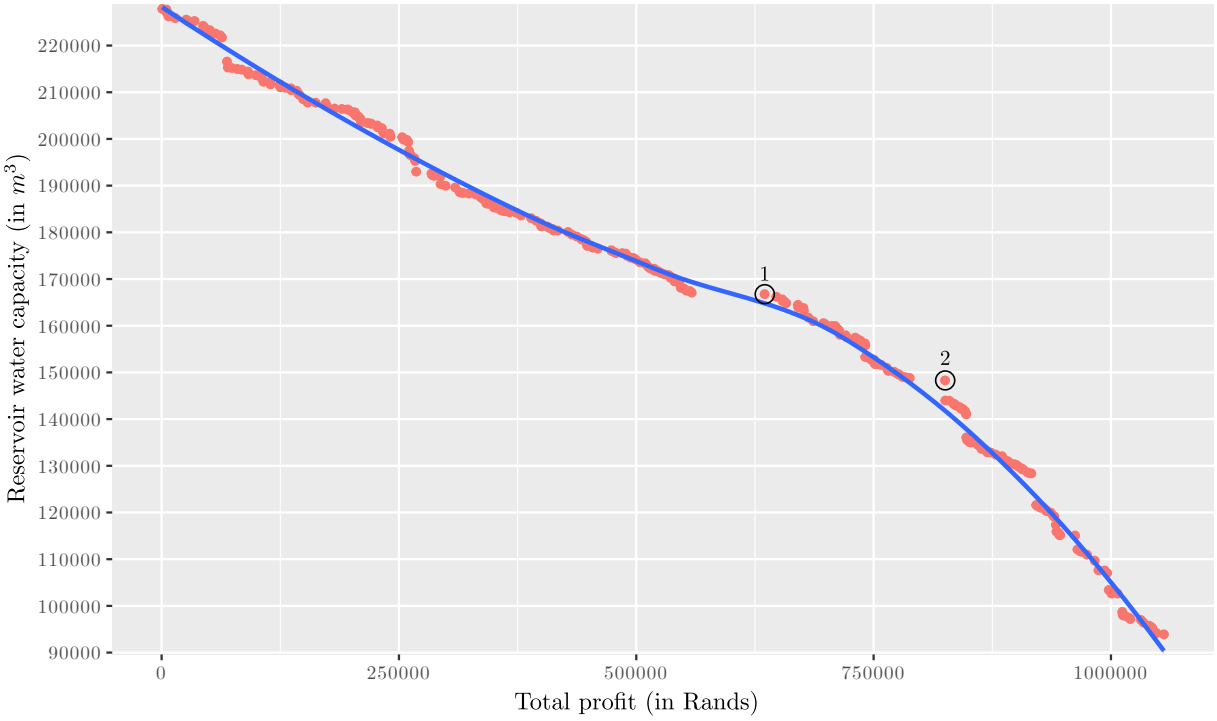


FIGURE 7.6: *The obtained Pareto optimal front when solving the bi-objective model (5.10)–(5.18) in the context of the case study for a lower amount of scheme water supply available.*

From the Pareto front of optimal solutions in Figure 7.6, a farmer may chose from a number of solutions that the model proposes. For each solution on the Pareto front, an irrigation and scheme water supply schedule exist, as described in the previous section, and may be selected by the farmer for implementation. The aim of the Pareto front is simply to give an overview of the multiple available implementations and upon selecting a solution(s) in the front, the exact key performance measures as well as an irrigation and scheme water supply schedule are given as output to the farmer.

If, for example, the farmer is more concerned with maximising his total profit rather than maximising the end-period reservoir water capacity, consider solutions 1 and 2 in Figure 7.6. Solution 1 achieves an approximate reservoir water capacity of 166 000 m^3 and an approximate total profit of R 656 250 while solution 2 achieves an approximate reservoir water capacity of 147 000 m^3 and an approximate total profit of R 890 000. For solution 1, an increase in the total

profit of approximately R 93 750 (R 656 250-R 562 500) is achieved for a small difference in the reservoir water capacity when compared to its nearest neighbour on the left, and for solution 2 an approximate increase in the total profit of R 77 500 (R 890 250-R 812 500) is achieved for a small difference in the reservoir water capacity when compared to its nearest neighbour on the left. For this reason, the farmer may rather consider to implement solution 1 or 2 rather than its nearest neighbour on the left, respectively, given that a large profit margin is obtained for a small difference in the end-period reservoir water capacity. Finally, it is also noted that for fewer amount of scheme water supply available, the range of the linear curve reduces, which results in a larger portion of the Pareto front to be visible.

7.2.3 A reflection on the results obtained

A reflection on the obtained results resulted in the following observations. It takes a considerably large amount of computational time (*i.e.* that is approximately two days) to produce solutions of good quality in the case of solving both mathematical models (5.2)–(5.9) and (5.10)–(5.18) for a large hypothetical scenario as described in §7.1. The reason for this is that during an iteration, a perturbation is only applied to one of three decision variables, respectively, with the hope that during the next iteration a perturbation is applied to a different decision variable. In this sense, there is some probability that the current decision variable will be again perturbed during the next iteration of the algorithm. There is, however, no risk associated with finding good solutions within a short time span when solving the mathematical models (5.2)–(5.9) and (5.10)–(5.18).

Furthermore, some indication of correlation between the obtained results in Figure 7.2, and scheduling scheme water supply during time periods where evaporation rates are low exist. The quality of the proposed scheme water supply schedule in Figure 7.2, however, may be improved. An example of such an improvement is to allocate more scheme water supply during time periods where evaporation rates are at its lowest (that may be during time periods 24–29). Both of the aforementioned observations calls for an updated neighbouring solution where the decision variables are simultaneously perturbed during a single algorithm iteration. This enables the algorithm to explore better combinations of decision variables in the solution vector more effectively. Moreover, the algorithm may also be executed for a larger number of i_{max} iterations to achieve this.

Finally, it is also important to mention that the bi-objective model (5.10)–(5.18) were executed for similar parameter values as the single-objective model (5.2)–(5.9). Considering the Pareto front, the obtained results seems of good quality given that few areas in the Pareto front are less populated. From Figure 7.6, it may be that the portion of the Pareto front for the total profit of approximately R 562 500 to R 656 250 for a reservoir water capacity of approximately 166 000 m^3 of water are not sufficiently explored. Regardless of this, it is save to assume that the parameter values proposed in §6.4 are sufficient to solve the bi-objective optimisation model (5.10)–(5.18).

7.3 Chapter summary

This chapter opened in §7.1 with a discussion on a hypothetical farm scenario developed as a case study. Next, in §7.2, the results obtained from solving the single-objective model (5.2)–(5.9) and bi-objective model (5.10)–(5.18) in the context of the hypothetical scenario were comprehensively discussed.

Considering the bi-objective optimisation model (5.10)–(5.18), it was found that a larger portion of the Pareto front is available for lower amounts of scheme water supply available. The resulting Pareto front is due to the non-linearity when reducing irrigation to crops. It was also found that the parameter values proposed in §6.4 as adopted in the bi-objective model (5.10)–(5.18) were sufficient. This chapter finally closed in §7.3 with a brief summary of the chapter contents.

CHAPTER 8

Decision support system

Contents

In this chapter, a computerised DSS is presented which is able to propose a water irrigation schedule and a scheme water supply schedule that may be used when water resources are limited. The irrigation and scheme water supply schedules are developed within the DSS with the aim to reflect a real life farming scenario.

This chapter opens in §8.1 with a short review on the basic notions involving DSSs from the literature. Next, in §8.2, the working of the DSS is thoroughly described by means of a walk trough of the *graphical user interface* (GUI) implemented in the model framework. Furthermore, the model framework is also discussed in some detail in this section by presenting a *data flow diagram* of the model framework. This is then followed by a discussion on two methods in §8.3 which may be used to employ and maintain the proposed DSS. Finally, this chapter closes in §8.4 with a summary of the chapter contents.

8.1 The basic notions in DSSs

A DSS may be defined as any computer-based *information system* that supports decision-making activities [102]. Another definition, according to Power [102], follows that a DSS may be described as an interactive computer-based system with the aim to aid in the decision making process. Some authors have extended the definition of a DSS and includes that any system which contributes towards decision making are considered as a DSS [117]. A DSS is, in essence, different to a traditional management information system in the sense that decision making support is emphasised in all its steps although the decision maker makes the final decision(s). Sprague [117], on the other hand, defined a DSS according to its characteristics as follows:

- traditional data access and retrieval functions are combined with mathematical models or analytic techniques,

- it integrates features that ease the operation of complexed models and analytic techniques for noncomputer users in an interactive way,
- it is more applicable to less well-structured, underspecified problems that managers often face, and
- it centres around adaptability and flexibility which allows for changes in the decision maker's approach, as well as a changing environment.

Moreover, Power [101] also defined three major characteristics of a DSS which included that a DSS is designed with the aim to assist in operations, must support decision making rather than automating it, and the system should be capable to respond quickly towards the changing needs and/or environment of the decision maker. Power [101] further defined five broad categories of DSSs, and included the following:

Communication-driven DSSs. These types of DSSs facilitates multiple users that works on a shared task where typical tasks include Google docs, Groove music, *etc.*

Data-driven DSSs. These type of DSSs are capable of accessing and manipulating a time series of internal data (these data may also be external at some times). Typical time series data include data warehousing, analytical systems, file drawer and managing reporting systems, executive information systems, and spatial DSSs. These DSSs are also frequently referred to as data-orientated DSSs.

Document-driven DSSs. These type of DSSs aims to retrieve, manage and manipulate unstructured information that manifest in a variety of electronic formats. Examples of such unstructured formats include catalogs, corporate records, corporate historical documents, policies and procedures, *etc.* These types of systems are also called knowledge management systems by some authors. The most prominent example of a powerful decision-making tool associated with a DSS is a search engine (*i.e.* Google, Firefox, *etc*).

Knowledge-driven DSSs. These types of DSSs communicates specialised problem solving expertise to the user in the form of facts, procedures and rules. A concept that relates to this is *data mining*.

Model-driven DSSs. These types of DSSs facilitates the access and manipulation of financial models, statistical models, optimisation models as well as simulation models. These types of systems utilises data and parameters provided by the user to support decision-making by analysing a scenario.

The primary focus of operations researchers lies within optimisation and simulation models as the "real" DSSs [102]. Power [102] further differentiates between enterprise-wide DSSs and desktop DSSs, where the prior is connected to a large data warehouse and serves multiple users, while the latter is a small system implemented on an individual user's PC [102]. The architecture of a DSS may also be divided into three fundamental components, known as the *database* (or *knowledge base*), the *model* (*i.e.* the decision factors and user criteria), and the *interface* presented to the user. The users themselves are also considered as a critical component in the architecture of a DSS [101].

The *user interface* (UI) is the space where interactions between humans and machines occur, and is sometimes referred to as a *machine interface* or a *human-computer interface* [78]. *Human-machine interaction* especially occurs in the field of industrial design and is often also referred to

as *human-computer interaction, computer-human interaction* or *man-machine interaction* [78]. Computer graphs, programming languages and operating systems are tools which may be configured by humans in the interface design of which the development thereof are typically based on the knowledge of computer science [78].

One of the most important types of UIs is the GUI. In modern times, the expression of a GUI is attributed to *human-machine interfaces* on computers since nearly all of them make use of graphics. Moreover, the term UI is generally associated with GUI due to increasing use of personal computers and the society's decreasing awareness of heavy machinery. In machinery control design and industrial control panel, the term *human-machine interface* is more regularly used [78]. In this thesis, a model-driven DSS is developed in conjunction with a desktop orientated DSSs which is configurable by the user.

8.2 The proposed DSS

The DSS developed in this thesis aims to incorporate both the single-objective optimisation model (5.2)–(5.9) and the bi-objective optimisation model (5.10)–(5.18), as described in Chapter 5, which is then solved in the context of a farm scenario using the adopted solution approaches described in Chapter 4. The DSS enable the decision makers (farmers) to present a realistic representation of their farming environment by uploading crop specific data, reservoir specific shape data and location specific evaporative A-pan historical data. A portion of the uploaded data may also be changed using a number of parameters within the DSS. In addition, the decision maker may also vary a number of performance parameters associated with the selected solution approach for improved results.

8.2.1 Graphical user interface design

The primary design requirement of a GUI is to facilitate the development of a tailored realistic farm scenario and then solve the optimisation model in the context of the realistic farm scenario by means of a solution approach. The decision maker, therefore, should be able to upload specific data into the DSS and select a solution approach to be employed in order to solve the mathematical optimisation models (5.2)–(5.9) and (5.10)–(5.18) in the context of the scenario. The results when solving the model is then given as output to the user.

Moreover, a package called *Shiny* [107], developed and supported by the software suite RStudio[108], is a powerful and elegant web framework which is typically used to generate reports and visualisations in RStudio with or without requiring web development skills [108]. This package is extremely useful when creating dynamic GUIs based on R script files and is, therefore, adopted in the development of this DSS.

The DSS demands a number of parameters as input in order to build a tailored realistic farm scenario. This entails uploading three templates onto the DSS, namely `crop_data`, `shape_data` and `evaporation_data` templates. The `crop_data` and `shape_data` includes the demand-related, farming-related and reservoir-related parameters as described in §5.2 whereas the `evaporation_data` contains the A-pan evaporation data gathered from a specific location also described in §5.2. Other parameters that are not included in the templates may be set during the execution of the DSS as part of the independent parameters in §5.2. It is important to note that only Excel Microsoft Office Open XML Format Spreadsheet files (`.xlsx`) or templates are allowed to be uploaded. Two examples of the templates as inputs to the DSS are graphically illustrated in Figures 8.1 and 8.2.

		Y_j^{avg}	p_j	l_j	v_j	d_j	O_{price}
		A	B	C	D	E	F	G
	1	Type	Average yield	Price per ton	Hectares	Variable cost	Fixed cost	Water cost
$j=1$	2	Maize	8	1520	200	2024	6133	3.9
$j=2$	3	Wheat	5.5	3473	25	3975	2789	3.9
$j=3$	4	Potato	32	574	45	1952	9213	3.9

(a) The average yield, production costs and the number of ha planted for crop j.

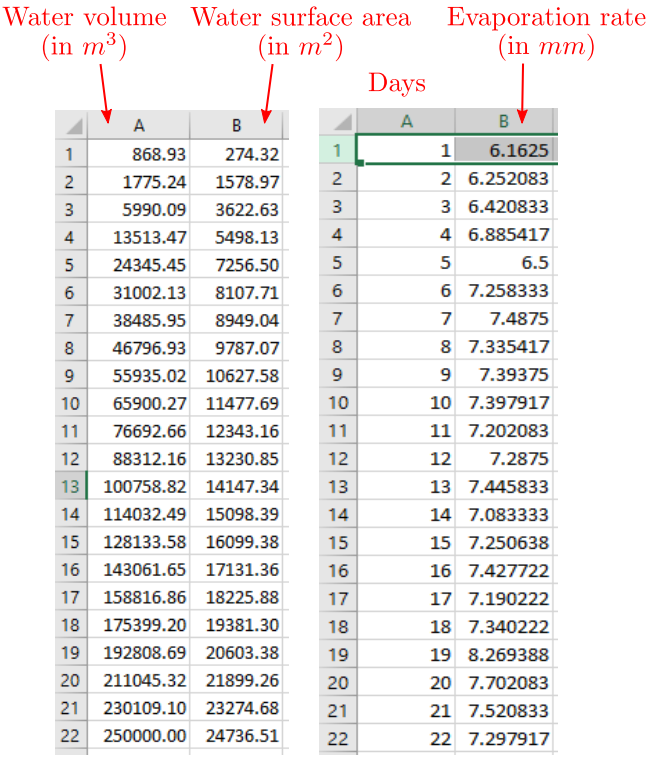
		A	B	C	D	E	F	G	H	
	1	Crop type	Paramete	1	2	3	4	5	6	Time period
$j=1$	2	Wheat	Stage	0	0	0	0	0	0	
	3	Wheat	Response	0	0	0	0	0	0	
	4	Wheat	Irrigation	0	0	0	0	0	0	
$j=2$	5	Maize	Stage	0	0	0	0	1	1	t
	6	Maize	Response	0	0	0	0	0.4	0.4	
	7	Maize	Irrigation	0	0	0	0	8.4	14.7	W_{jt}
$j=3$	8	Potato	Stage	4	4	4	0	0	0	
	9	Potato	Response	0.3	0.3	0.3	0	0	0	K_{jt}
	10	Potato	Irrigation	32.3	30.8	6.2	0	0	0	

(b) The time of growth, yield response factors and crop water demands for crop j.

FIGURE 8.1: *An example of the* `crop_data.xlsx` *file containing a number of parameters which may be changed according by the decision maker within the respective files. The appended symbols in red correspond to the parameters defined in Chapter 5.*

In Figure 8.1(a), the average yield for crop j, the cost of producing yield from crop j and the number of hectares planted for crop j are contained within the file. These parameters are essential when estimating the total profit. The symbols highlighted in red illustrates how the columns in the input `.xlsx` file corresponds to the parameters defined in Chapter 5. From Figure 8.1(b), the crop growth stages, the yield response factors and the crop water requirement for crop j, respectively, during each time period are contained within the file. The symbols highlighted in red illustrates how the rows in the input `.xlsx` file corresponds to the parameters defined in Chapter 5. It is important to note that the format of the input `.xlsx` files (Figure 8.1(a) – (b)) may not change — this may result in a shift in the columns or removing columns. Moreover, Figure 8.2(a) illustrates the shape characteristic data which relates the reservoir water volume with reservoir water surface area, and Figure 8.2(b) illustrates the average historical evaporation rate or each day in a hydrological year. The format of the input `.xlsx` files (Figure 8.2(a) – (b)) may also not change.

Suppose that the decision maker wants to add or remove data contained in the template files, consider for example Figure 8.2 where data is removed from the template. Given that potatoes are not currently grown on his/her farm, row 4 in Figure 8.1(a) and rows 8–10 in Figure 8.1(b) may be removed from the templates. In order to add data to these templates, a row is merely added with the respective information corresponding to each column in Figure 8.1(a), whereas three rows are added with the respective information corresponding to each column in Figure 8.1(b). Moreover, suppose that the decision maker wants to manipulate the shape characteristic data or evaporation data. For the shape characteristic date in Figure 8.2(a), the decision maker may simply replace the row and columns values in Figure 8.2(a) with the new shape data, independent of the length of the data (*i.e.* length of column). For the evaporation data in Figure 8.2(b), the decision maker may simply adjust the evaporation rate (that is the second column in Figure 8.1(b)).



	A	B
1	868.93	274.32
2	1775.24	1578.97
3	5990.09	3622.63
4	13513.47	5498.13
5	24345.45	7256.50
6	31002.13	8107.71
7	38485.95	8949.04
8	46796.93	9787.07
9	55935.02	10627.58
10	65900.27	11477.69
11	76692.66	12343.16
12	88312.16	13230.85
13	100758.82	14147.34
14	114032.49	15098.39
15	128133.58	16099.38
16	143061.65	17131.36
17	158816.86	18225.88
18	175399.20	19381.30
19	192808.69	20603.38
20	211045.32	21899.26
21	230109.10	23274.68
22	250000.00	24736.51

(a) The reservoir water volume and corresponding water surface area.

	A	B
1	1	6.1625
2	2	6.252083
3	3	6.420833
4	4	6.885417
5	5	6.5
6	6	7.258333
7	7	7.4875
8	8	7.335417
9	9	7.39375
10	10	7.397917
11	11	7.202083
12	12	7.2875
13	13	7.445833
14	14	7.083333
15	15	7.250638
16	16	7.427722
17	17	7.190222
18	18	7.340222
19	19	8.269388
20	20	7.702083
21	21	7.520833
22	22	7.297917

(b) The average daily A-pan evaporation rate.

FIGURE 8.2: *An example of the file containing reservoir specific and A-pan historical data. In (a), the reservoir water volume with its corresponding water surface area are listed. In (b), the historical A-pan evaporation rate for each day in a year are listed.*

Upon initiating the DSS, the decision maker is introduced to a welcome screen where instructions are provided for navigation purposes (the introduction tab/page are not graphically illustrated). The decision maker is then prompted to select a solution approach that is employed — the single-objective optimisation model (5.2)–(5.9) may be solved by means of the SA algorithm whereas the bi-objective optimisation model (5.10)–(5.18) may be solved by means of the DBMOSA, respectively. Next, the user may navigate to the following tab/page, called the "Load data" tab/page, by selecting the continue button. A graphical illustration of the "Load data" tab/page is shown in Figure 8.3.

On the left hand side in Figure 8.3, the three templates/files (Figure (8.1) and (8.2)) are conventionally uploaded onto the DSS *via* the use of *Shiny*'s `fileInput` fields. Once the files has been uploaded (as shown in Figure 8.3 by the blue bar beneath the `fileInput` fields on the left hand side), data tables and graphs are computed which illustrates the data that was currently uploaded — the decision maker may validate the uploaded data before continuing to the next tab/page. In the bottom left corner in Figure 8.3, the decision maker may continue onto the next page or return to the previous tab/page to alter or change some of the information fields. From Figure 8.3, it is clear that only the parameters confined to maize were inserted into the `.xlsx` files, where the corresponding information is illustrated in a table format. The data table on the top displays the parameters contained in the `.xlsx` file in Figure 8.1(a) whereas the second data table displays the parameters contained in the `.xlsx` file in Figure 8.1(b).

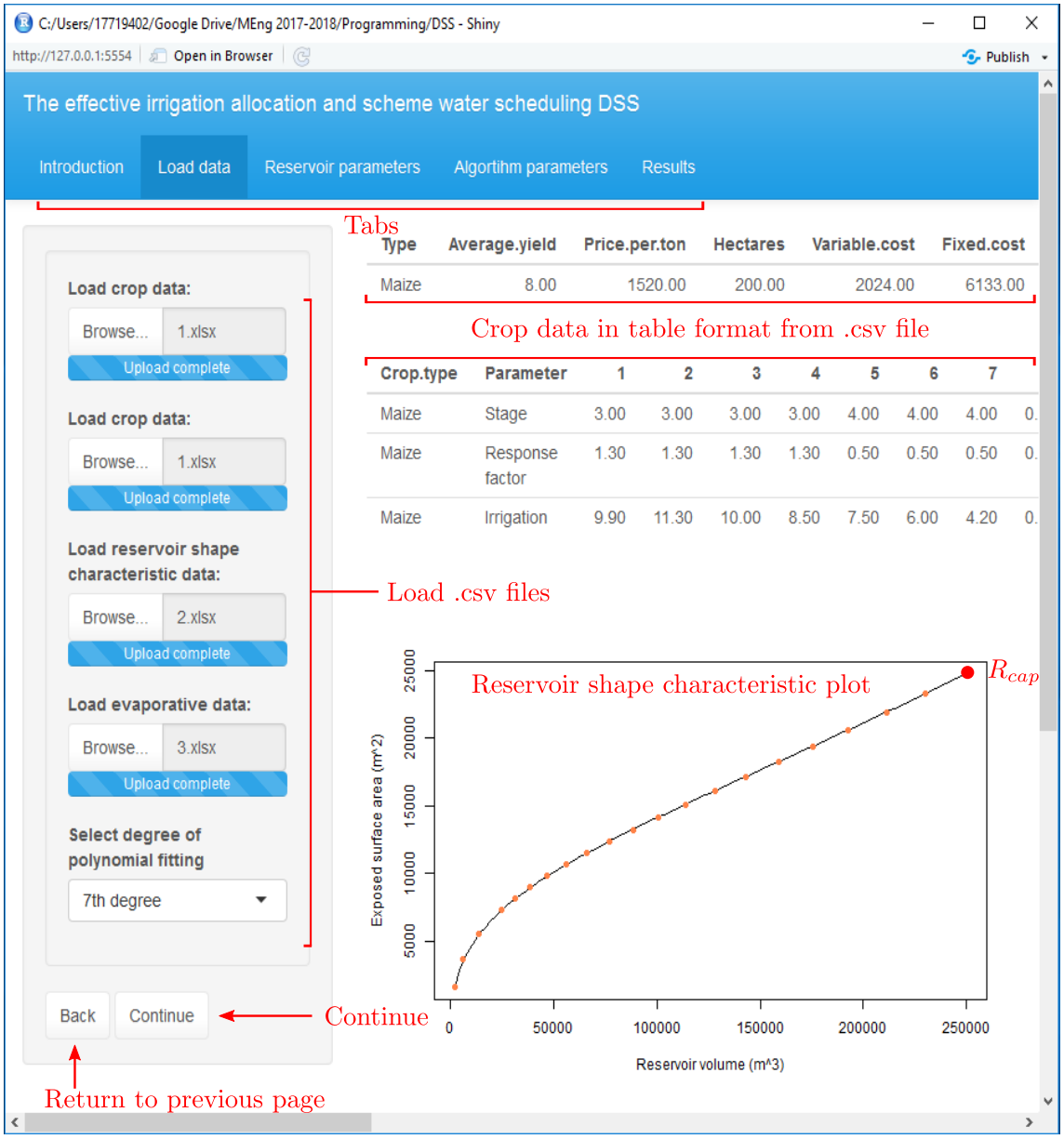


FIGURE 8.3: *A screenshot (with appended descriptions in red of the proposed irrigation allocation and scheme water supply DSS) of the* `fileInput` *fields with corresponding output fields generated from the "Load data" tab/page. This page focus on uploading* `.xlsx` *files onto the DSS.*

The piecewise linear approximation fitted to the reservoir shape characteristic data in Figure 8.2(a) and the R_{cap} (*i.e.* the total reservoir water capacity) are also illustrated in Figure 8.3. From this data, the R_{cap} is located and taken as input to the model. It is also possible to change the degree of fit applied the average daily A-pan evaporation data — this is not shown in Figure 8.3. The initial degree of fit is taken as a 7^{th} degree as explained in §6.1. Once the decision maker validated the uploaded data, he or she may navigate to the next tab/page by selecting the "Continue" button on the bottom left corner in Figure 8.3. Upon selecting this button, the data that was entered into the respective `fileInput` fields are stored. When returning to tab/pages, the tab panel on the top or the "Return" button may be used in Figure 8.3.

The "Reservoir parameters" tab/page is illustrated in Figure 8.4 and enables the decision maker to select a number of values associated with the reservoir activities, and include the current reser-

voir water capacity and the end-period reservoir water capacity. Two types of *Shiny* `widgets` are found in this figure which are used to change parameter values. The widgets include *Shiny*'s `sliderInput` element and *Shiny*'s `numericInput` element. The `sliderInput` element is conventionally used when the parameter in question has a desired range associated with it (minimum and maximum allowed/suggested values), and that the range for this parameter may not be violated. In this case, initial or default values are populated in these user input fields, and may be noticed when selecting this tab/page at first. A data table on the right hand side in Figure 8.4 illustrates the default values of the input fields, and is reactively changed once a change in any of the input fields are made.

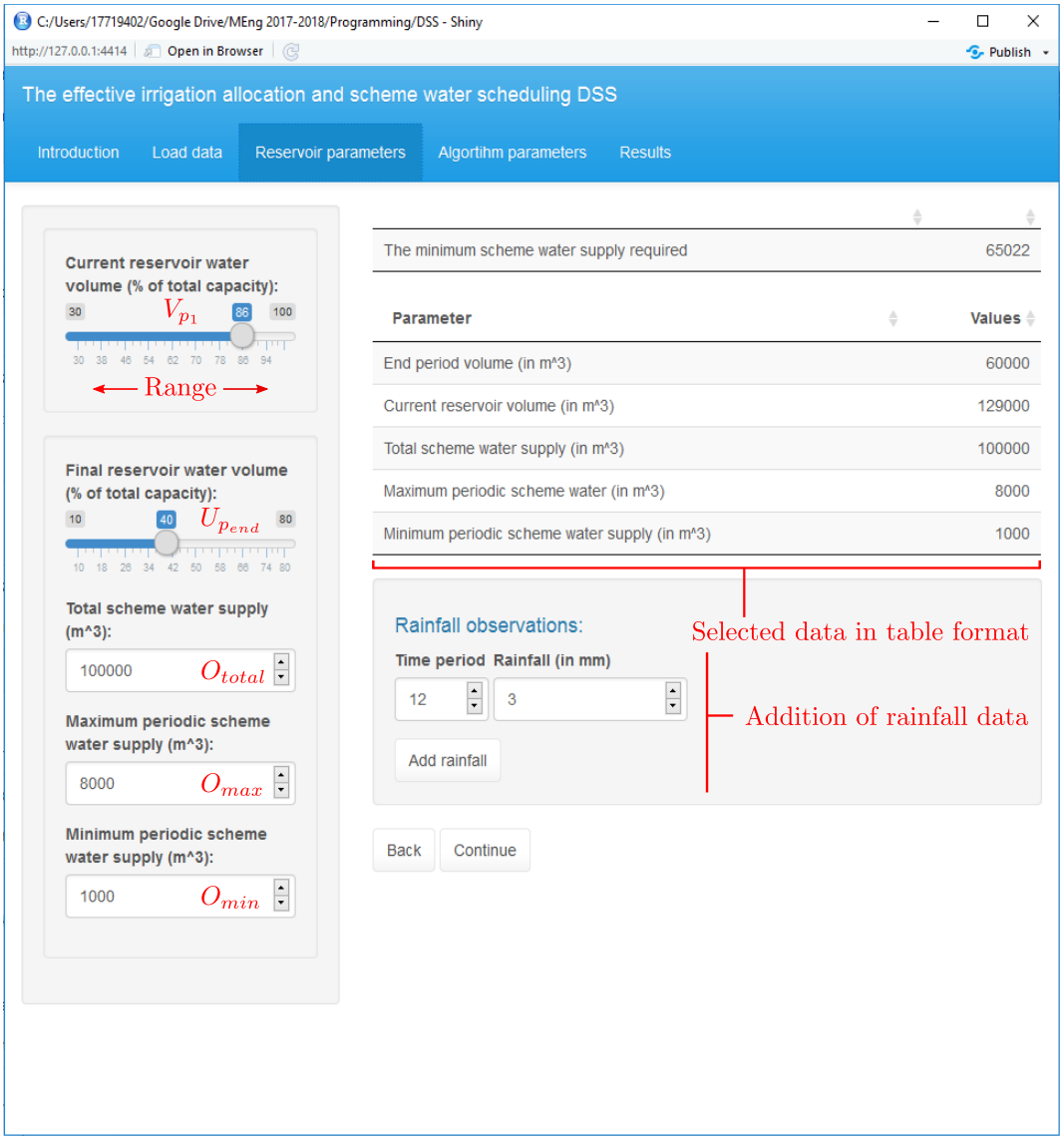


FIGURE 8.4: *A screenshot (with appended descriptions in red of the proposed irrigation allocation and scheme water supply DSS) of the "Reservoir parameters" tab/page where five parameter values associated with reservoir water activities may be adjusted. In addition, rainfall measures may also be added according to each time period.*

In Figure 8.4, the first `sliderInput` element (in the top left corner of Figure 8.4) is the current reservoir water volume (% of total capacity), the second `sliderInput` element (below the first `sliderInput` element) denotes the final reservoir water volume (% of total capacity), the third

`numericInput` element is the total scheme water supply (in m^3), the fourth `numericInput` element is the maximum periodic scheme water supply (in m^3), and the fifth `numericInput` element is the minimum periodic scheme water supply (in m^3). Moreover, in Figure 8.4, the top column in the main panel indicates the minimum scheme water supply that is required within the `numericInput` element called "Total scheme water supply (m^3)" for the scenario to be feasible. Depending on the type of optimisation selected in the "Introduction" tab/page, the `sliderInput` element called "Final reservoir water volume (% of total capacity)" are adjusted, meaning that for the bi-objective model (5.10)–(5.18) the `sliderInput` element is removed since this input is not required by this particular method. The decision maker may also add additional rainfall measures, and in order to do so, two `numericInput` elements are used as illustrated in Figure 8.4. The decision maker may therefore select a specific time period and enter the amount of rainfall (in millimetres) for the respective time period, and then submit the data by selecting the "Add rainfall" button.

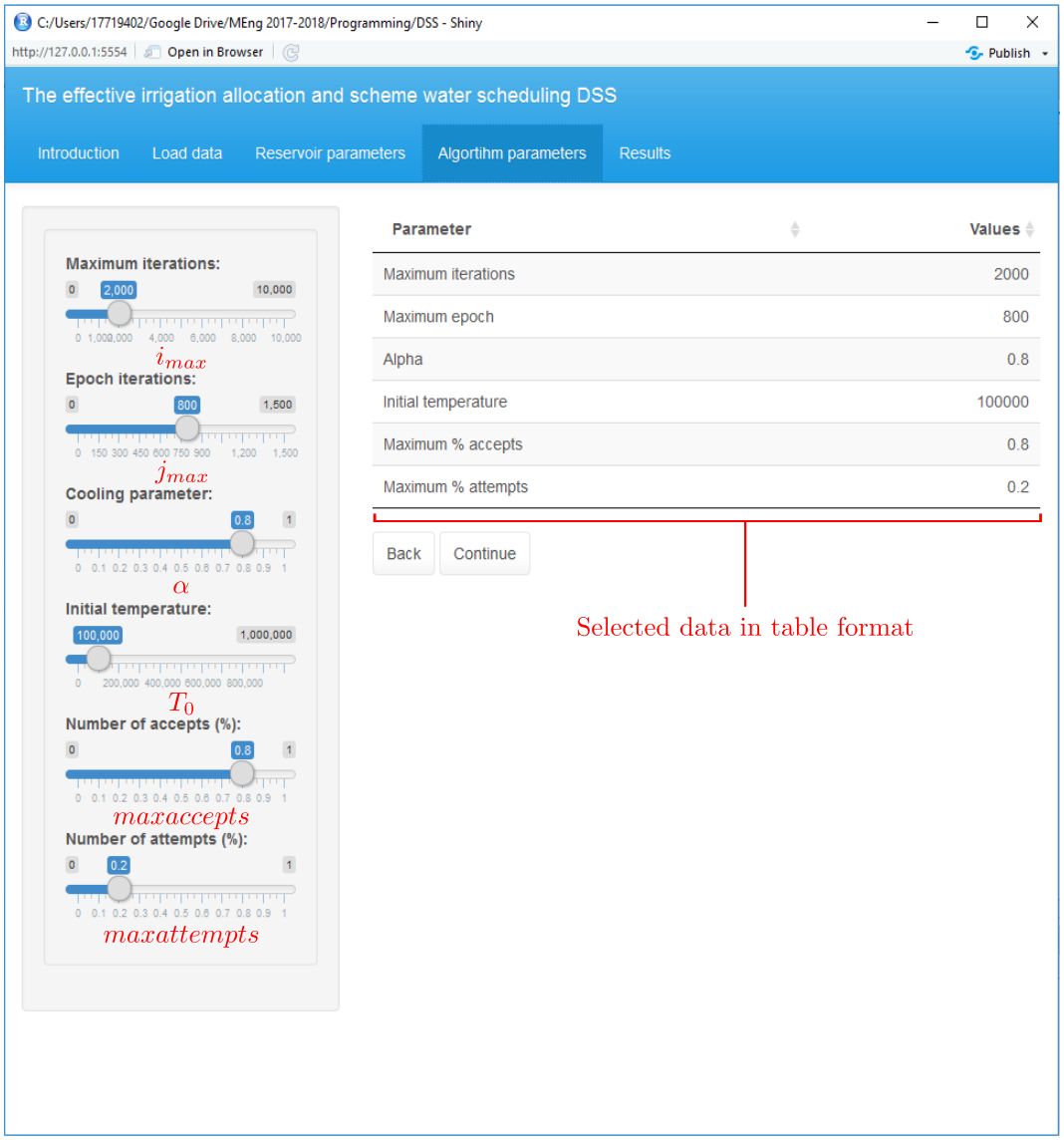


FIGURE 8.5: *A screenshot (with appended descriptions in red of the proposed irrigation allocation and scheme water supply DSS) of the "Algorithm parameters" tab/page where six performance parameters associated with the type of method selected for optimisation in the "Introduction" tab/page.*

The "Algorithm parameters" tab/page is illustrated in Figure 8.5 where the decision maker may adjust six performance parameters associated with the selected solution approach (*i.e.* solving the single-objective optimisation model using the SA algorithm or solving the multi-objective optimisation model using DBMOSA) in the home tab/page. All the parameters associated with the selected solution approach are adjusted using *Shiny*'s `sliderInput` element. If, however, the decision maker is not familiar with these parameters, the user may leave these fields unchanged as it is already pre-set according to default values obtained by the algorithmic parameter evaluation in §6.3. The pre-set values are shown in Figure 8.5. Similar to as in the "Reservoir parameters" tab/page, a data table is displayed showing the default values of each `sliderInput` element on the left hand side, and is adjusted once a change is made in any of the input fields.

The first `sliderInput` element in the top-left hand side of Figure 8.5 denotes the maximum number of iterations i_{max} that the algorithm should execute, the second `sliderInput` element denotes the maximum number of epoch iterations, known as *maxepoch*, that should be executed, the third `sliderInput` element denotes the cooling parameter α associated with the cooling schedule, the fourth `sliderInput` element denotes the initial temperature T_0, the fifth `sliderInput` element denotes the maximum accepts, known as *maxaccepts*, as a % of the maximum number of epoch iterations, and the sixth `sliderInput` element denotes the maximum attempts, known as *maxattempts*, as a % of the maximum number of epoch iterations. The symbols highlighted in red in Figure 8.5 corresponds with the parameters defined in §4.3.

Upon selecting the single-objective optimisation model (5.2)–(5.9) from the "Introduction" tab/page, the results obtained when solving this model in the context of the farm scenario are illustrated in Figure 8.6. The "Run" button in the top-left hand corner in Figure 8.6 may be selected in order to execute the algorithm, and after the algorithm has finished, the "Download results" button may be selected in order to download the results. The results are then stored in a `.xlsx` file to a preferred location on the desktop computer. A progress bar on the bottom right-hand corner is also implemented which illustrates the progression of the algorithm, as illustrated in Figure 8.6 in the bottom right hand corner. Once the algorithm has terminated, the progress bar disappears and the results are illustrated in the tab/page as in Figure 8.6.

In the *key performance measures* table, the estimated profit (in Rands) is shown in conjunction with the end-period reservoir water capacity (in m^3), the total scheme water cost (in Rands), and the total evaporation (in m^3) taking place over the scheduling horizon, as shown in Figure 8.6. The *scheme water schedule* table is shown below the key performance measures table where the time periods Z_p of scheme water supply are shown with the corresponding amount of scheme water supply O_p during this period. Finally, the irrigation schedule is illustrated by means of a bar chart below the scheme water supply schedule, where on the x-axis the time period is shown and on the y-axis the amount of water (in m^3) to be irrigated to the respective crop type is shown. Each respective crop is indicated by a colour where the colour legend are located on the right-hand side in Figure 8.6. The way in which this table should be interpretation is comprehensively described in the previous section.

In Figure 8.7, the obtained Pareto front when solving the bi-objective model (5.10)–(5.18) in the context of the hypothetical farm scenario are graphically illustrated as output to the user. In the Pareto front, the x-axis denotes the estimated profit (in Rands) and the the y-axis denotes the end-period reservoir water capacity (in m^3). Moreover, the red dots in the Pareto front denotes the approximate Pareto optimal solutions whereas the blue line denotes a regression fitted to the approximate Pareto optimal solutions. From the Pareto front, a *user brush* may be employed, as illustrated in Figure 8.7, to select a range of solutions on the front where the corresponding key performance measures are then displayed below the plot in a table format. From this table, the decision maker may change the number of entries (at the top left hand side

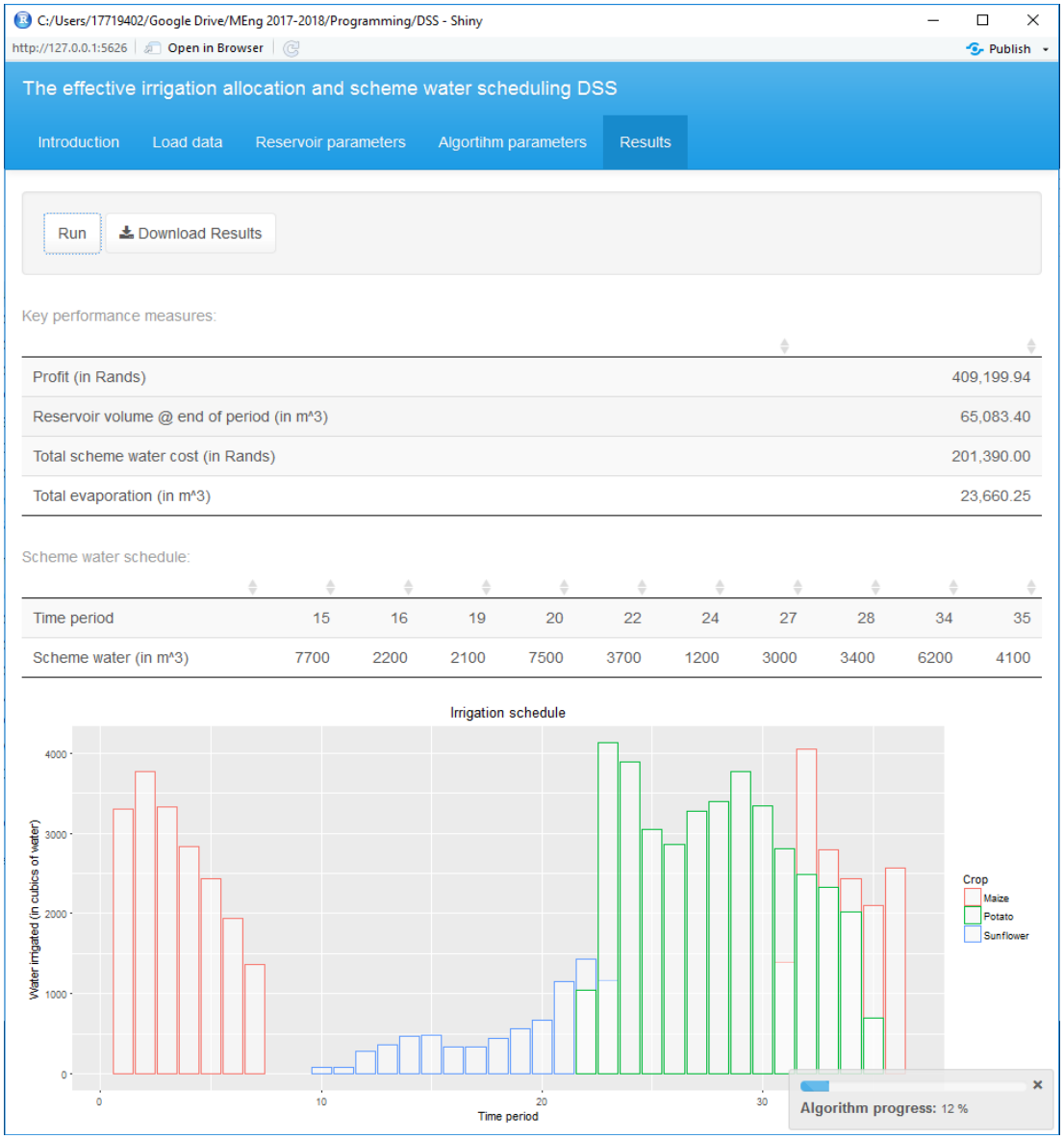


FIGURE 8.6: *A screenshot of the proposed irrigation allocation and scheme water supply DSS of the "Results" tab/page where the results for the SA algorithm are illustrated (selecting single-objective optimisation). The results obtained may also be downloaded in* `.xlsx` *file from the "Download results" button.*

of the data table in Figure 8.7), search for specific entries among the selected entries using a numeric value in the search box (at the top right hand side of the data table in Figure 8.7), or page through the number of solution selected by the user brush (on the bottom right hand side corner of the data table in Figure 8.7).

8.2.2 DSS model framework

In this section, the logical back end working of the DSS are elucidated according the a top-down approach to diagramming data movement using *data flow diagrams* (DFDs), as described by Kendall and Kendall [71]. Incorporating this approach enables a pictorial depiction of the data processes involved in the DSS and helps to conceptualise how data flows during the execution

FIGURE 8.7: *A screenshot of the proposed irrigation allocation and scheme water supply DSS of the "Results" tab/page where the results for the DBMOSA (bi-objective optimisation) are illustrated. The obtained results may also be downloaded in* `.xlsx` *file from the "Download results" button.*

of the DSS [71]. From the DFD, a number of external entities are indicated for the purpose of showing where the required information may be gathered, as well as the role that some of the entities plays in the working of the DSS. The Diagram 0[1] of the model framework incorporated in the DSS is illustrated in Figure 8.8. From this figure, four external entities are visible, namely the South African Weather Service, CropWat 8.0, FAOSTAT website and the decision maker, as well as six processes and two workspaces.

According to Kendall and Kendall [71], an entity is defined as a department, business, or a machine that can send or receive data from the system, and thus, the aforementioned entities are classified as such. Furthermore, entities are denoted by a double square and a process is

[1]This diagram is an explosion of the context diagram which provides an overview of the general system by considering only inputs and outputs and a single process [71].

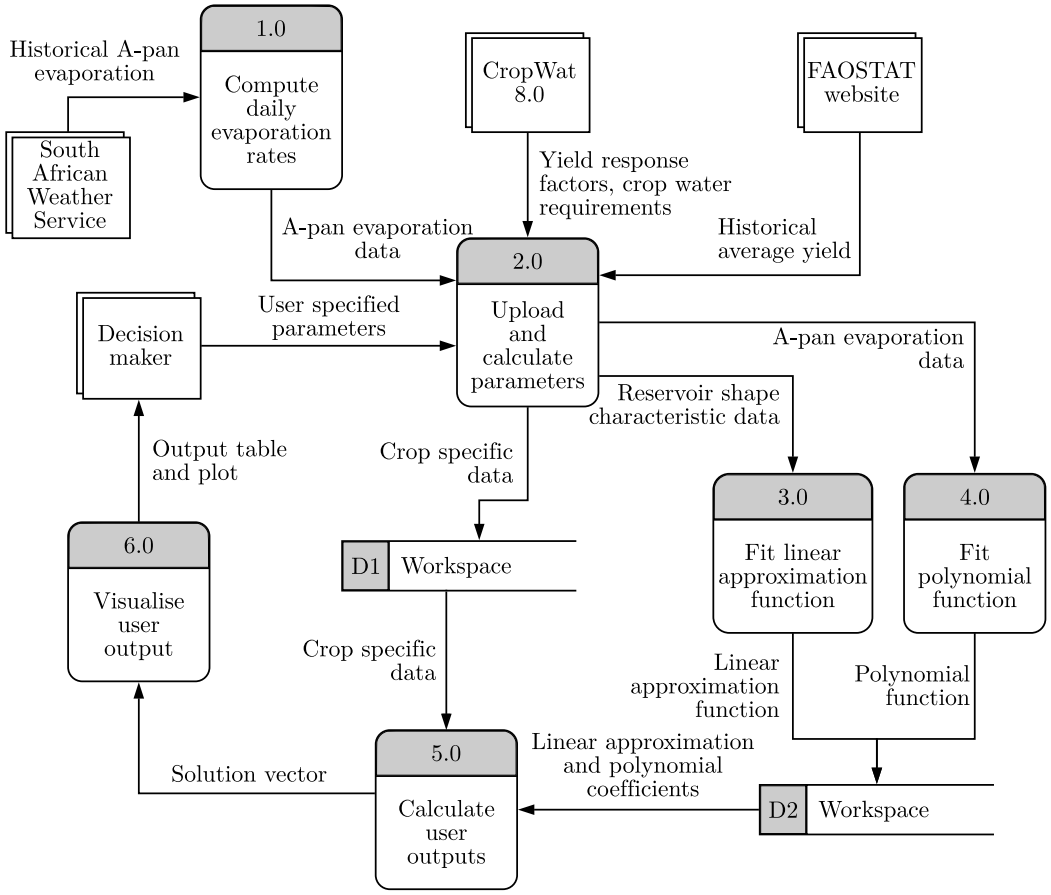


FIGURE 8.8: *The diagram 0 of the working of the DSS illustrating the gathering of data from external entities, the user inputs and the corresponding outputs generated by the DSS.*

denoted by a rectangle with rounded corners. A process is responsible for transforming data, hence, the data that leaves the process is always classified differently than the data that enters the process. Finally, a data store is denoted by an open-end rectangle and simply shows a depository or retrieval of data [71]. In the remainder of this section, each process in the Diagram 0 is discussed with some detail with respect to the notions that was discussed in Chapters 2, 4, 5 and 6.

In Figure 8.8, process 1.0 is responsible for gathering, cleaning and computing the A-pan evaporation data gathered from the South African Weather Service for a selected location. The cleaning of the data refers to altering the data such that it is useable, whereas computing refers to calculating the A-pan daily historical average evaporation rate by using the historical data from a number of years, as gathered from the South African Weather Service. After the data has been cleaned and the historical daily A-pan evaporation rates has been calculated, the calculated rates may be inserted into the `evap_data` template (as shown in Figure 8.2(b)) which is then uploaded onto the DSS as explained in the previous section.

Process 2.0 in Figure 8.8 is responsible for gathering crop-related, demand-related and reservoir-related parameters provided by the external entities called CropWat 8.0 and the FAOSTAT website, as well as gathering independent parameters such as the targeted end-period reservoir water capacity and the historical daily A-pan evaporation data, as calculated by process 1.0. The way in which the required data are gathered from the aforementioned external entities are briefly described in §2.4.3 and §2.4.6 whereas the type of data that is uploaded by process

2.0 are elucidated in §5.2. The data is gathered by uploading the templates files illustrated in Figures 8.1 and 8.2 to the respective `fileInput` fields using the GUI, as explained in §8.2.1. The decision maker (farmer) also specifies a number of parameters using the GUI which is also uploaded to the system. From the uploaded data, crop specific parameters are computed and stored in workspace D1 whereas the remaining uploaded data are altered for processes 3.0 and 4.0. It is important to note that the historical average yield obtained for a crop(s) may be manually inserted into the `.xlsx` file in Figure 8.1(a) as a user-input.

After the relevant data are uploaded onto the system, process 3.0 is responsible for fitting a linear approximation to the reservoir shape characteristic data, as mentioned in §2.3 and done in §6.1. The coefficients of the linear approximation function are then stored in the workspace called D2. In a similar fashion, process 4.0 is responsible for fitting a polynomial function to the historical daily A-pan evaporation data of which the coefficients for the polynomial fitted function are also stored in workspace D2. In Figure 8.3 from §8.2.1, the degree of the polynomial fit may be changed according to the decision maker where the new polynomial coefficients for the new fit are then updated in D2.

Process 5.0 is responsible to calculate the user outputs in the form of an optimal solution vector, as shown in Figure 6.3 in §6.2.1, and is done by incorporating the SA algorithm or the DBMOSA, as described in §4.3.1 and §4.3.3, to solve the single-objective model (5.2)–(5.9) or the bi-objective model (5.10)–(5.18). Process 5.0 takes as input the crop specific parameters from workspace D1 and then collect coefficient of the linear approximation and polynomial function from workspace D2 in order to calculate the end-period reservoir water capacity, as described in §5.6. The four key performance measures (*i.e.* the total profit, end-period reservoir water capacity, total scheme water cost and total evaporation) as well as the irrigation and scheme water supply schedule are captured in the solution vector after processes 5.0 has completed, of which the solution is then passed to process 6.0 in the case of solving the single-objective model (5.2)–(5.9). In the case where the bi-objective model (5.10)–(5.18) was selected, an archive which contains the nondominated set of solutions as well as the corresponding solution vector for each nondominated solution is then passed to processes 6.0.

Finally, process 6.0 is responsible for visualising the resulting solution vector or archive of solutions received from process 5.0 such that it is easily interpretable. In the case of solving the single-objective model (5.2)–(5.9), the four key performance measures and the scheme water supply schedule are visualised in a table format, as illustrated in Figure 8.6, whereas the irrigation schedule is visualised using a bar chart, as shown in Figure 8.6. In the case of solving the bi-objective model (5.10)–(5.18), a Pareto front of optimal solutions is visualised, as illustrated in Figure 8.7, where multiple solutions may be selected using a user brush for implementation purposes.

8.3 DSS deployment and maintenance

Multiple platforms exist over which the DSS described in §8.2.1 may be deployed in order for users to run it on their computers [106]. For this thesis, the following two avenues are suggested for deployment and maintenance:

Accessible over the web. In this case, the only requirement is the need of a web browser. This may be beneficial to nontechnical computer users in the sense that no program installation is required. The application that was developed may be hosted through two platforms, namely the *Shiny Server* program or the *Shinyapps.io* hosting service (RStudio's

hosting service). These services are usually free of charge for a specific trial period, after which an of money is charged once the trial expires [78].

Run locally. In this case, the user is required to have R and *Shiny* installed on their computer. The code required for the application may be made available by means of two methods, namely hosting it online (*i.e.* in a GitHub repository) or as a (.zip) file which is personally shared with the user. The prior is considered more desirable in the sense that maintenance and updates may be performed in a remote fashion by the developer (shown as an "update"), which is downloadable through R. The latter, however, entails manually sharing (either email or shared) the updated application with the user, of which the user is then required to unzip and replace the existing files in the current working directory with the newly updated files.

8.4 Chapter summary

This chapter was devoted to a comprehensive discussion on the working of the DSS. The chapter opened in §8.1 with a short review on the basic notions found in the literature based on DSSs. A model-driven desktop oriented DSS was developed in this thesis. Next, in §8.2, the working of the DSS was comprehensively described by using screenshots from the GUI, as implemented in the model framework. This was followed by a brief description of the model framework using a diagram 0 as part of a DFD where each process was discussed in some detail. In §8.3, two methods were briefly mentioned on how to deploy and maintain DSSs. Finally, this chapter closed in §8.4 with a summary of the chapter contents.

CHAPTER 9

Conclusion

Contents

This summative chapter comprises out of three sections. The chapter opens in §9.1 with a detailed summary of the contents in this thesis in a chapter-by-chapter overview fashion. Next, four novel contributions of this thesis are highlighted in §9.2. This chapter is then closed in §9.3 with five avenues for suggestions of further investigations as possible follow-up work on the four novel contributions achieved in this thesis.

9.1 Thesis summary

In the first chapter of this thesis, the reader was introduced to the affects that global warming and the El Niño phenomenon have on the availability of water resources as well as the affect of these phenomenons on farmers with respect to crop yield loss. These natural phenomena provided the context for the informal problem description that was considered in this thesis. The problem description, therefore, proposed the development of a DSS for farmers to make effective decisions with respect to water management during times where water availability is limited. The thesis objectives and scope was also outlined in Chapter 1, and the chapter closed with a thesis organisation in the final section.

Chapter 2 opened with a short summary on irrigation and the research of development of irrigation methods in the United States of America. This was followed by a thorough discussion on evaporation and transpiration from soil and crops, as well as estimating evaporation from reservoir water surface areas. As part of estimating crop final yield using CWPFs, the role of crop growth stages in estimating crop water requirements and the role that crop coefficients and yield response factors play were comprehensively described. Next, the accuracy of CWPFs when estimating the final crop yield were statistically analysed of which the most accurate CWPF were selected for implementation in this thesis. This led to a thorough discussion on similar crop production decision support systems where the working of CropWat 8.0 as part of a decision support system were discussed thoroughly. CropWat 8.0 plays an essential role when estimating the crop coefficients and yield response factors in CWPFs. The components of soil moisture management systems were also briefly discussed, and as conclusion to this chapter, the

limitations associated with the aforementioned crop coefficient approach of estimating final crop yield were discussed comprehensively.

In Chapter 3, a discussion on computational complexity was given with regards to solving optimisation problems. Next, the mathematical formulation in general form of MOOPs were presented. The importance of convexity and non-convexity in MOOPs were also documented, and led to the conclusion that in general non-convex MOOPs are considerably harder to solve that convex MOOPs. Next, the literature review shifted towards the notions of solution dominance in MOOPs of which the properties thereof and the concepts of Pareto optimality were thoroughly described. This was followed by a brief description on methods to determine nondominated sets of solutions using the properties of solution dominance. As required by some solution methodologies, multiple nondominated sets of solutions are required of which the method of FNSA were briefly discussed. Finally, the chapter closed with a discussion on the simplest of methods to solve MOOPs called the weighted-sum of objectives method.

Chapter 4 was devoted to a review of the literature based on three classes of solution methodologies when solving optimisation problems. The first class is exact solution approaches of which an exact solution is obtained when solving an optimisation problem, and entailed a thorough description on the branch-and-bound method proposed by Doig and Land [77] and the method of total enumeration as examples in this class. The second class is heuristic solution approaches of which an approximate solution are computed rather than an exact solution. In this class, the quality of the solution obtained is sacrificed for the benefit of computational time. The third and final class is metaheuristic solution approaches which is different in nature compared to the aforementioned class in the sense that these solution approaches are able to escape from local optima. Within this class, two optimisation paradigms were considered namely trajectory-based and population-based approaches. For each paradigm, an example of such a solution approach was reviewed and included the following: The SA algorithm proposed by Kirkpatrick *et al.* [73] and the genetic algorithm proposed by Holland [63] for single-objective optimisation, and the DBMOSA proposed by Smith *et al.* [115] and the NSGA-II proposed by Agarwal *et al.* [28] for multi-objective optimisation. For each optimisation environment (that is for the single-objective and multi-objective optimisation environment), a trajectory-based and population-based paradigm were considered as a solution methodology.

In Chapter 5, two novel mathematical models were presented to the reader and formulated based on the proposed modelling framework considered in this thesis. The proposed models are similar in nature to one another, however, solved in different optimisation environments (that is a single-objective or bi-objective environment). As introduction, assumptions related to the behaviour of an open-air irrigation reservoir on which the mathematical models were based on were thoroughly discussed. This was followed by a brief description on the proposed modelling framework that is considered in this thesis as well as multiple input parameters to the proposed mathematical models. Next, two approaches that are related to rainfall and the prediction thereof as input to the reservoir were comprehensively discussed. It was found that due to the variability of rainfall predictions, it carries less risk to formulate an irrigation and scheme water schedule assuming that no rainfall will take place. Moreover, if rainfall occurred, this may be entered as an input to the model of which the model is then re-executed for an updated reservoir water capacity. Next, the two novel mathematical models were discussed comprehensively, and this was followed by a brief discussion on computing reservoir periodic end-volumes as well as the end-period reservoir water capacity as part of the final section in this chapter.

Chapter 6 was devoted to elucidate on the model implementations when solving a hypothetical scenario, and to validate the two mathematical models proposed in Chapter 5. First, a small hypothetical instance was formulated, proposed and solved using the aforementioned mathematical

models. Next, the model implementations to solve the hypothetical instance were thoroughly discussed. Next, an evaluation was done on the algorithmic parameters to compute a good set of parameter values for sufficient algorithmic results. The results obtained from solving the hypothetical instance was then used to validate both the aforementioned mathematical models by means of three different validation techniques, namely a face validation, a random benchmark validation and consulting an expert in the field of crop irrigation and farming. As part of the validation techniques is the consulting of an expert in the field, of which the expert concluded that the proposed models as well as the obtained results, the estimated crop parameters, and the proposed modelling framework were realistic, authentic and naturally correct.

In Chapter 7, a realistic case study was presented to the reader and solved using both the mathematical models proposed in Chapter 5. The results obtained from both the mathematical models when solving the realistic case study was comprehensively discussed assuming that no rainfall will take place. This resulted in an optimal irrigation and scheme water supply schedule for the single-objective optimisation (5.2)–(5.9) and a Pareto front for the bi-objective optimisation model (5.10)–(5.18).

In Chapter 8, the working of the DSS was thoroughly discussed. This chapter opened with the basic notions found within DSSs, and yielded that a model-driven desktop oriented DSS was proposed in this thesis. Next, the working of the DSS was thoroughly described by using a GUI design, and followed by a comprehensive description on the implemented model framework. The chapter then closed with a discussion on the deployment steps and maintenance that should included in the development of DSSs.

9.2 Appraisal of thesis contributions

Four main contributions were made in this thesis. In this section, these contributions are elucidated and briefly described.

Contribution 1: *A comprehensive review on the literature pertaining to agricultural prerequisites related to predicting crop yield from water deficits, obtaining parameter values for the adopted approach, and a review of crop production decision support systems.*

A complete and comprehensive literature review is presented in this thesis related to the agricultural prerequisites for irrigating crops when considering reservoir water inflows and outflows, water deficits in crop growth stages as well as crop production decision support systems. A number of CWPFS proposed by two different authors are statistically analysed. From these results, a CWPF were formulated by combining the approach of Doorenbos and Kassam [35] and a multiplicative-type of model proposed by Stewart *et al.* [121] to use in the objective function of the models developed later in this thesis. Obtaining specific parameter values related to the aforementioned formulated CWPF were also considered in such a manner that it is easy implementable by any user. Some of the parameters are obtained from crop production decision support systems which yields an in-depth description on CropWat 8.0 and the working thereof.

Moreover, the working of soil moisture management systems and the role that these systems may play in the CWPF formulated were also briefly described. This also entails a short description on some of the components within such a system. A brief conclusion regarding the limitations associated with the adopted crop coefficient approach are also elucidated. This provides some background on essential factors to consider within the adopted approach.

Contribution 2: *A novel single-objective and bi-objective crop irrigation and scheme water supply model.*

Another contribution of this thesis is the formulation of two novel mathematical models, namely a single-objective crop irrigation and scheme water supply model (5.2)–(5.9) and a bi-objective crop irrigation and scheme water supply model (5.10)–(5.18). According to the best knowledge of the author, no reference in the literature could be found which simultaneously schedule irrigation to crops that are grown while proposing a scheme water schedule to acquire additional water resources at some cost. Georgiou and Papamichail [54] attempted a similar problem where a non-linear programming model was formulated and incorporates a soil-water balance equation to determine optimal reservoir release strategies and the optimal cropping pattern for irrigated crops while considering various weather conditions. Their model considered various probability levels for rainfall, evapotranspiration and inflows to the reservoir to estimate the optimal distribution area, water irrigated to crops and the total farm income.

Similar to the approach of Georgiou and Papamichail [54], the proposed mathematical models (5.2)–(5.9) and (5.10)–(5.18) estimates the farm income from crop yield as a result of irrigation, while simultaneously considering the reservoir water inflows and outflows. The proposed models, however, are different in nature to the models proposed by Georgiou and Papamichail [54] given that the inflow of the reservoir related to scheme water supply are taken as a decision variable in order to propose an effective schedule that considers the amount of water lost due to evaporation. As part of estimating the actual yield, a crop coefficient approach is undertaken rather than a soil water balance equation as was done by Georgiou and Papamichail [54]. Furthermore, an irrigation and scheme water schedule are developed based on the assumption that no rainfall occurs which shifts the focus of this thesis towards providing decision support to farmers in times where water supply are limited. Multiple solutions for different reservoir water volumes are also computed in the form of a Pareto front for a broader context related to multiple implementable solutions.

Contribution 3: *A demonstration of the practical working of the novel models formulated in Contribution 2 within the context of a realistic hypothetical farm scenario formulated as a case study.*

A small hypothetical problem instance representing a realistic scenario was developed in order to illustrate the workability and practicality of the proposed novel mathematical models (5.2)–(5.9) and (5.10)–(5.18). The instance involves a farm located in the Bredarsdorp region where location specific evaporation data and crop parameters where gathered for the growing of maize. The numerical results obtained from solving the hypothetical instance are used as part of validation purposes for both the mathematical models. Consulting with an expert in the field of crop irrigation and farming, it was found that a similar approach is adopted when developing an irrigation system that incorporates crop coefficients when computing crop water requirements for such a system. Moreover, another hypothetical scenario was formulated as part of a case study and solved using the two novel mathematical models (5.2)–(5.9) and (5.10)–(5.18). In this scenario, three different crops were considered and grown in the Swartland district area. These crops included wheat, maize and potatoes.

Contribution 4: *A generic decision support system which incorporates the novel single-objective model (5.2)–(5.9) and the novel bi-objective model (5.10)–(5.18) and is capable of providing decision support to farmers when water supply is limited.*

The main contribution of this thesis is a generic computerised DSS which is capable of solving any farm scenario formulated within the DSS that is exposed to an environment where water supply is limited. The two novel mathematical models (5.2)–(5.9) and (5.10)–(5.18) presented

in Contribution 2 are incorporated in order to provide decision support to farmers with respect to the farming scenario. Furthermore, a farmer may utilise this DSS in order to assist with the decision making with respect to water management during water restricted times by considering and analysing the suggested irrigation and scheme water supply schedules provided by the DSS.

Contribution 5: *Suggesting a number of ideas for possible future work following on the contributions in this thesis.*

The last contribution of this thesis is put forward in the next section. The suggestions that are made aims to help orientate research related to agricultural perquisites as part of promoting sustainable development within this sector. The documented avenues may be further investigated as possible follow-up work to the contributions in this thesis.

9.3 Future work

In this section, final suggestions are provided to the reader for future work related to a number of aspects in this theses. The following two suggestions are made relating to the working of the single-objective and bi-objective optimisation models (5.2)–(5.9) and (5.10)–(5.18) and the validation of these models:

Suggestion 1: *Revise Algorithms 6.2 and 6.3 for generating neighbouring solutions.*

In §7.2, based of a reflection on the results obtained, it was emphasised that an updated neighbouring solution may result in a computational benefit when solving similar problems of the same complexity and size. The need for simultaneously perturbing the decision variables $\left(\frac{w}{ET}\right)_{jt}$ and O_p during an epoch iteration (that is a small perturbation to the solution vector) may provide the algorithm with the opportunity to compare the benefit when doing so and therefore, may be able to provide better quality neighbouring solutions faster. Currently, only a single decision variable are perturbed iteratively. Moreover, the solution quality may also be affected by this as more combinations of the decision variables in the solution vector are considered iteratively.

Suggestion 2: *Adopt a full factorial design in the parameter evaluation for optimal algorithmic results.*

The parameter evaluation experiment conducted in §6.3 did not consider a full factorial design, since excessive amounts of computational time is required to solve such an experiment when adopted for a similar problem as the proposed hypothetical small instance in §6.1. Only a number of algorithmic parameter combinations were tested as part of the experiment due to the computational time involved in executing the experiment. It may be that the optimal algorithmic parameter combination were not considered in the parameter evaluation conducted in §6.1. Therefore, a full factorial design may be implemented to find the optimal combination of algorithmic parameter values for best algorithmic results assuming that computational time is not limited.

The following two suggestions are related to a scope enlargement of this thesis:

Suggestion 3: *Take into account different irrigation methods and the efficiency of these methods.*

A number of crops may be irrigated using different irrigation methods. In this thesis, the method used to irrigate crops is not taken into consideration as a result of increased model complexity. As part of a scope enlargement, different irrigation methods may be considered as well as the efficiency thereof when irrigating crops in order to propose an irrigation schedule that considers the crop water requirements as well as the irrigation efficiency of the irrigation method. One

way to implement this is to select an irrigation method for a respective crop type and then specify the irrigation efficiency of this method as it is typically known in advance. This may also act as an input to the proposed DSS.

Suggestion 4: *Formulate an effective scheme water supply schedule for an environment where water supply is not limited.*

This thesis specifically focus on proposing an irrigation and scheme water supply schedule when water supply is limited for more effective use of water resources during times of hardship. As part of a scope enlargement, a scheme water supply schedule may be formulated which minimises the total evaporation from reservoir water surface areas while ensuring that crop water requirements are met for maximum crop yield production in an environment where water supply is not limited.

The following suggestion is related to the natural paradigm of this thesis:

Suggestion 5: *Consider other methods for modelling the effect of water deficits on crop yield.*

As discussed in §2.6, the yield response factor approach adopted in this thesis are limited to the accuracy of the crop yield response factors adopted when computing final crop yield. Another approach to consider when estimating the affect of water deficits on crop yield is the soil water balance equation as adopted by Georgiou and Papamichail [54].

References

[1] Abdullah S, Golafshan L, Zakree M & Nazri A, 2011, *Re-heat simulated annealing algorithm for rough set attribute reduction*, **6(8)**, pp. 2083–2089.

[2] Allen RG, Pereira LS, Raes D & Smith M, 1999, *Crop evapotranspriation: Guidelines for computing crop water requirements*, Irrigation and Drainager Paper No. 56, FAO, Rome, Italy.

[3] American Meteorological Society, 2012, *Meteorology Glossary: Precipitation*, [Online], [Cited June 2018], Available from `http://glossary.ametsoc.org/wiki/Precipitation`.

[4] Back T, 1996, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, Oxford university press, Oxford.

[5] Barrett JP, 1974, *The coefficient of determination — some limitations*, The American Statistician, **28(1)**, pp. 19–20.

[6] Bartram C, 2018, *Top 10 reasons we need to drink water*, [Online; posted 27-July-2018], url: `https://www.bodybuilding.com/content/top-10-reasons-we-need-to-drink-water.html`.

[7] Bertsimas D & Tsitsiklis J, 1993, *Simulated annealing*, Statistical science, **8(1)**, pp. 10–15.

[8] Borchers HW, *pracma: Practical Numerical Math Functions*, R package version 2.1.8, 2018, url: `https://CRAN.R-project.org/package=pracma`.

[9] Branke J, Deb K & Miettinen K, 2008, *Multiobjective optimization: Interactive and evolutionary approaches*, Springer, New York (NY).

[10] Bras RL & Cordova JR, 1981, *Intraseasonal water allocation in deficit irrigation*, Water resources research, **17(4)**, pp. 866–874.

[11] Briggs LJ & Shantz H, *The water requirements of plants: 1. Investigations in the Great Plains in 1919 and 1911*, US Bureau of Plant Industries, Bulletin No. 24, Washington (DC).

[12] Briggs LJ, 1914, *Relative water requirements of plants*, Journal Agriculture Research, **3**, pp. 1–63.

[13] Brown P, 2014, *Basics of evaporation and evapotranspiration*, College of Agriculture and Life Sciences, University of Arizona, Tucson (AZ).

[14] Brownlee J, 2011, *Clever algorithms: Nature-inspired programming recipes*, LuLu: Jason Brownlee, Melbourne.

[15] Busetti F, 2003, *Simulated Annealing Overview*, [Online], [Cited May 2018], Available from `http://www.%20aiinfinance.%20com/saweb.%20pdf`.

[16] Campbell G & Diaz R, *Simplified soil-water balance models to predict crop transpiration*, ICRISAT, Patancheru, India.

[17] Coello CC, 2006, *Evolutionary multi-objective optimization: A historical view of the field*, IEEE computational intelligence magazine, **1(1)**, pp. 28–36.

[18] Coello CAC, Lamont GB & Van Veldhuizen DA, 2007, *Evolutionary algorithms for solving multi-objective problems*, Springer, New York (NY).

[19] Computus Management Information (Pty) Ltd, 2010, *Enterprise budgets 2009/2010*, [Online], [Cited May 2018], Available from `http://www.computus.co.za/Publikasies/Vertakkingsbegrotings09Engels.pdf`.

[20] Corwin DL, Rhoades JD & Simunek J, 2007, *Leaching requirement for soil salinity control: Steady-state versus transient models*, Agricultural Water Management, **90(3)**, pp. 165–180.

[21] De Jager JM, 1994, *Accuracy of vegetation evaporation ratio formulae for estimating final wheat yield*, Water SA, **20(4)**, pp. 307–314.

[22] De Jager JM, Potgieter AB & Van den Berg WJ, 1998, *Framework for forecasting the extent and severity of drought in maize in the Free State Province of South Africa*, Agricultural Systems, **57(3)**, pp. 351–365.

[23] De Jager JM, 1968, *Carbon dioxide exchange and photosynthetic activity in forage grasses*, PhD thesis, University College of Wales, Aberystwyth.

[24] De Jager J, Van Zyl W, Kelbe B & Singels A, 1987, *Research on a weather service for scheduling the irrigation of winter wheat in the Orange Free State region*, Final Research Report to the Water Research Commission by the Department of Agrometeorology, University of Free State, WRC Report No. 177/1/87.

[25] De Wit CT, 1958, *Transpiration and crop yields*, Agriultural Research Reports, **64(6)**, pp. 1–88.

[26] Deb K, 2001, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, New York (NY).

[27] Deb K, 1999, *Multi-objective genetic algorithms: Problem difficulties and construction of test problems*, Evolutionary computation, **7(3)**, pp. 205–230.

[28] Deb K, Agrawal S, Pratap A & Meyarivan T, 2000, *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, Proceedings of the International Conference on Parallel Problem Solving From Nature, pp. 44–53.

[29] Delamarre D & Virot B, 1998, *Simulated annealing algorithm: technical improvements*, RAIRO-Operations Research, **32(1)**, pp. 43–73.

[30] Department of Agriculture, Forestry and Fisheries, 2016, *Production guideline for wheat*, (Unpublished) Technical Report.

[31] Department of Agriculture, Forestry and Fisheries, 2016, *Trends in the agricultural sector*, (Unpublished) Technical Report.

[32] Department of Agriculture, Forestry and Fisheries, 2017, *Trends in the agricultural sector*, (Unpublished) Technical Report.

[33] Department of Communication and Information System, 2016/17, *South Africa Yearbook*, (Unpublished) Technical Report.

[34] Department of Water Affairs and Sanitation, 2018, *DWS*, (Unpublished) Technical Report.

[35] Doorenbos J & Kassam AH, 1979, *Yield response to water*, Irrigation and Drainager Paper No. 33, FAO, Rome, Italy.

[36] Doorenbos J & Pruit WO, 1975, *Guidelines for predicting crop water requirements*, Irrigation and Drainager Paper No. 24, FAO, Rome, Italy.

[37] Dorigo M, 1992, *Optimization, learning and natural algorithms*, Ph.D. Thesis, Politecnico di Milano, Milan.

[38] Dréo J, Pétrowski A, Siarry P & Taillard E, 2006, *Metaheuristics for hard optimization: Methods and case studies*, Springer, Berlin.

[39] Du Plessis J, 2003, *Maize production in South Africa*, Directorate of Agricultural Information Services, Department of Agriculture, Pretoria, South Africa.

[40] Eakin HM & Brown CB, 1939, *Silting of reservoirs*, US Department of Agriculture, Washington DC.

[41] Earth Institute, 2016, *El Nino and global warming – what's the connection?*, [Online], [Cited February 2018], Available from `https://phys.org/news/2016-02-el-nino-global-warmingwhat.html`.

[42] Earth Science Office, 1999, *Soil moisture*, [Online], [Cited June 2018], Available from `https://weather.msfc.nasa.gov/`.

[43] Engrand P, 1998, *A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management*, Proceedings of the 5^{th} International Conference on Nuclear Engineering, pp. 416–423.

[44] Epstein D, 2016, *How accurate are long-range weather forecasts?*, [Online; posted 15-June-2016], url: `https://www.boston.com/weather/weather/2016/06/15/just-accurate-long-range-forecast-anyway`.

[45] Erasmus D, 2012, *Growing maize in the Swartland*, [Online; posted 3-March-2012], url: `https://www.farmersweekly.co.za/crops/field-crops/growing-maize-in-the-swartland/`.

[46] ETH Zurich: Institute for Operations Research, 2018, *Combinatorial Optimization*, [Online], [Cited October 2018], Available from `https://www.math.ethz.ch/ifor/research/combinatorial-optimization.html`.

[47] Fogel DB, 2000, *What is evolutionary computation?*, IEEE spectrum, **37(2)**, pp. 26–32.

[48] Food and Agriculture Organization of the United Nations, 2018, *Aquacrop*, [Online], [Cited February 2018], Available from `http://www.fao.org/aquacrop/en/`.

[49] Food and Agriculture Organization of the United Nations, 2018, *ClimWat 2.0*, [Online], [Cited May 2018], Available from `http://www.fao.org/land-water/databases-and-software/climwat-for-cropwat/en/`.

[50] Food and Agriculture Organization of the United Nations, 2018, *CropWat 8.0*, [Online], [Cited February 2018], Available from `http://www.fao.org/land-water/databases-and-software/cropwat/en/`.

[51] Food and Agriculture Organization of the United Nations, 2018, *FAOSTAT*, [Online], [Cited February 2018], Available from `http://www.fao.org/faostat/en/#home`.

[52] Food, Organization A, Heng L, Food, of the United Nations AO, Moutonnet P & Smith M, 2002, *Deficit Irrigation Practices*, Food and Agriculture Organization of the United Nations, Rome, Italy.

[53] Geem ZW, Kim JH & Loganathan GV, 2001, *A new heuristic optimization algorithm: Harmony search*, Simulation, **76(2)**, pp. 60–68.

[54] Georgiou PE & Papamichail DM, 2008, *Optimization model of an irrigation reservoir for water allocation and crop planning under various weather conditions*, Irrigation Science, **26(6)**, pp. 487–504.

[55] Gerber R, 2018, BEng (Civil) and MSAII (Designer) at *Breerivier Irrigation*, [Personal Communication], Contactable at `rg@breebesp.co.za`.

[56] Glover F, 1986, *Future paths for integer programming and links to AI*, Computers and Operations Research, **13**, pp. 553–549.

[57] Goldberg D, 1989, *Genetic algorithms in optimization, search and machine learning*, Reading, Addison-Wesley, New York (NY).

[58] Grain SA, 2018, *Safex Feeds*, [Online], [Cited May 2017], Available from `http://www.grainsa.co.za/pages/industry-reports/safex-feeds`.

[59] Hanks R, 1983, *Yield and water-use relationships: An overview*, pp. 393–411 in Taylor H (Ed), *Limitations to efficient water use in crop production*, American Society of Agronomy, Madison, WI.

[60] Henning MA & Van Vuuren JH, *Graph and network theory*, In process.

[61] Hillier FS & Lieberman GJ, 2004, *Introduction to operations research*, McGraw-Hill, New York (NY).

[62] Hogg RV & Tanis EA, 2009, *Probability and statistical inference*, Prentice Hall: Pearson, New Jersey (NJ).

[63] Holland JH, 1992, *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*, MIT Press, Cambridge.

[64] Hollander M, Wolfe DA & Chicken E, 2014, *Nonparametric statistical methods*, 3rd Edition, John Wiley & Sons, Hoboken (NJ).

[65] HydraWize, 2018, *HydraWize: Soil moisture management*, [Online], [Cited June 2016], Available from `http://hydrawize.com/`.

[66] Igbadun HE, Tarimo AK, Salim BA & Mahoo HF, 2007, *Evaluation of selected crop water production functions for an irrigated maize crop*, Agricultural Water Management, **94(1-3)**, pp. 1–10.

[67] Investopedia, 2018, *Gross Domestic Product (GDP)*, [Online], [Cited June 2018], Available from `https://www.investopedia.com/terms/g/gdp.asp`.

[68] Irrigation Education, 2016, *How a Center Pivot Irrigation Machine Works*, [Online], [Cited January 2018], Available from `http://blog.irrigation.education/blog/how-a-center-pivot-works`.

[69] Jensen M, 1968, *Water consumption by agricultural plants*, Academic Press, New York & Londen, **2**, pp. 1–22.

[70] Katz J, Menezes AJ, Van Oorschot PC & Vanstone SA, 1996, *Handbook of applied cryptography*, CRC press, New York (NY).

[71] Kendall KE & Kendall JE, 2011, *Systems analysis and design*, 8th Edition, Pearson Education Limited, Edinburgh, England.

[72] Kipkorir EC, Raes D & Massawe B, 2002, *Seasonal water production functions and yield response factors for maize and onion in Perkerra, Kenya*, Agricultural Water Management, **56(3)**, pp. 229–240.

[73] Kirkpatrick S, Gelatt CD & Vecchi MP, 1983, *Optimization by Simulated Annealing*, Science, **220(4598)**, pp. 671–680.

[74] Kramer O, 2017, *Genetic algorithm essentials*, Springer, New York (NY).

[75] Kung HT, Luccio F & Preparata FP, 1975, *On finding the maxima of a set of vectors*, Journal of the ACM, **22(4)**, pp. 469–476.

[76] Laerd Statistics, 2018, *Wilcoxon Signed-Rank Test using SPSS Statistics*, [Online], [Cited October 2018], Available from `https://statistics.laerd.com/spss-tutorials/wilcoxon-signed-rank-test-using-spss-statistics.php`.

[77] Land AH & Doig AG, 1960, *An automatic method of solving discrete programming problems*, Econometrica: Journal of the Econometric Society, **29**, pp. 497–520.

[78] Lindner BG, 2017, *Bi-objective generator maintenance scheduling for a national power utility*, PhD thesis, Stellenbosch University, Stellenbosch.

[79] Lötter DP, 2017, *Design of a weapon assignment subsystem within a ground-based air defence environment*, PhD thesis, Stellenbosch University, Stellenbosch.

[80] MacMillan A, 2016, *Everything you wanted to know about our changing climate but were too afraid to ask*, [Online], [Cited March 2018], Available from `https://nrcca.cals.cornell.edu/soil/CA3/CA0324.php`.

[81] Makeham JP & Malcolm L, 1986, *The economics of tropical farm management*, Cambridge University Press, Cambridge.

[82] Malkawi AIH & Al-Sheriadeh M, 2000, *Evaluation and rehabilitation of dam seepage problems. A case study: Kafrein dam*, Engineering Geology, **56(3-4)**, pp. 335–345.

[83] Marica A, 2004, *Short description of CROPWAT model*, [Online], [Cited July 2017], Available from `http://agromet-cost.bo.ibimet.cnr.it/fileadmin/cost718/repository/cropwat.pdf`.

[84] Meza GR, Ferragud XB, Saez JS & Durá JMH, 2016, *Controller tuning with evolutionary multiobjective optimization: A holistic multiobjective optimization design procedure*, Springer International Publishing.

[85] Miettinen K, 1999, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Dordrecht.

[86] Minhas BS, Parikh KS & Srinivasan TN, 1974, *Toward the structure of a production function for wheat yields with dated inputs of irrigation water*, Water Resources Research, **10(3)**, pp. 383–393.

[87] Mladenović N & Hansen P, 1997, *Variable neighborhood search*, Computers & operations research, **24(11)**, pp. 1097–1100.

[88] Moran J, 2009, *Business management for tropical dairy farmers*, CSIRO publishing, Collingwood.

[89] National Climate Assessment, 2017, *Extreme Weather*, [Online], [Cited May 2014], Available from `https://nca2014.globalchange.gov/highlights/report-findings/extreme-weather`.

[90] News 24, 2017, *SA highly affected by global warming CSIR*, [Online], [Cited April 2014], Available from `http://www.news24.com/Green/News/SA-highly-affected-by-global-warming-CSIR-20140408`.

[91] Northeast Region Certified Crop Adviser Study Resources, 2017, *Competency Area 3: Drainage and irrigation AEM*, [Online], [Cited September 2010], Available from `https://nrcca.cals.cornell.edu/soil/CA3/CA0324.php`.

[92] Nourani Y & Andresen B, 1998, *A comparison of simulated annealing cooling strategies*, Journal of Physics A: Mathematical and General, **31(41)**, pp. 8373.

[93] NSW Government: Department of Primary Industries, 2016, *Plant nutrients in the soil*, (Unpublished) Technical Report.

[94] Panda RK, Behera SK & Kashyap PS, 2003, *Effective management of irrigation water for wheat under stressed conditions*, Agricultural water management, **63(1)**, pp. 37–56.

[95] Parks GT, 1996, *Multiobjective pressurized water reactor reload core design by nondominated genetic algorithm search*, Nuclear Science and Engineering, **124(1)**, pp. 178–187.

[96] Parsopoulos KE & Vrahatis MN, 2002, *Particle swarm optimization method in multiobjective problems*, Proceedings of the 2002 ACM symposium on Applied computing, pp. 603–607.

[97] Petrowski JD & Taillard PSE, 2006, *Metaheuristics for hard optimization*, Springer, New York (NY).

[98] Poli R, Kennedy J & Blackwell T, 2007, *Particle swarm optimization*, Swarm intelligence, **1(1)**, pp. 33–57.

[99] Potatoes South Arica, 2018, *Multimedia Guide to Potato Production in South Africa*, [Online], [Cited October 2018], Available from `http://www.potatoes.co.za/research/potato-guidelines.aspx`.

[100] Potatoes South Arica, 2018, *Potatoes South Africa*, [Online], [Cited October 2018], Available from `http://www.potatoes.co.za/`.

[101] Power DJ, 2002, *Decision support systems: concepts and resources for managers*, Quorum Books, Westport (CT).

[102] Power D, 1997, *What is a DSS*, The Online executive journal for data-intensive decision support, **1(3)**, pp. 223–232.

[103] Rardin RL & Rardin RL, 1998, *Optimization in operations research*, Prentice Hall, New Jersey (NJ).

[104] Rhenals AE & Bras RL, 1981, *The irrigation scheduling problem and evapotranspiration uncertainty*, Water Resources Research, **17(5)**, pp. 1328–1338.

[105] Romeo F, Sangiovanni VA & Huang M, 1986, *An efficient general cooling schedule for simulated annealing*, Proceedings of the IEEE International Conference on Computer Aided Design, pp. 381–384.

[106] RStudio, 2018, *RStudio – Sharing apps to run locally*, [Online], [Cited October 2014], Available from `https://shiny.rstudio.com/articles/deployment-local.html`.

[107] RStudio Inc, *Easy web applications in R*, URL: `http://www.rstudio.com/shiny/`, 2013.

[108] RStudio Team, *RStudio: Integrated Development Environment for R*, RStudio, Inc, Boston, MA, 2015, url: `http://www.rstudio.com/`.

[109] Sastry K, 2005, *Genetic algorithms*, pp. 97–125 in Burke EK & Kendall G (Eds), *Search methodologies*, Springer, New York (NY).

[110] Schlunz EB, 2016, *Multiobjective in-core fuel management optimisation for nuclear research reactors*, PhD thesis, Stellenbosch: Stellenbosch University.

[111] Schulze RE, 1974, *Catchment evapotranspiration in the Natal Drakensberg*, PhD thesis, University of Natal, Pietermaritzburg.

[112] Seeds for Africa, 2018, *Western Cape Vegetable Planting Chart — South Africa*, [Online], [Cited October 2018], Available from `https://www.seedsforafrica.co.za/pages/western-cape-vegetable-planting-chart-1`.

[113] Sen S & Sherali HD, 1985, *A branch and bound algorithm for extreme point mathematical programming problems*, Discrete applied mathematics, **11(3)**, pp. 265–280.

[114] Singels A, Annandale J, Jager JD, Schulze R, Inman-Bamber N, Durand W, Rensburg LV, Heerden PV, Crosby C, Green G & Steyn J, 2010, *Modelling crop growth and crop water relations in South Africa: Past achievements and lessons for the future*, South African Journal of Plant and Soil, **27(1)**, pp. 49–65.

[115] Smith KI, Everson RM, Fieldsend JE, Murphy C & Misra R, 2008, *Dominance-Based Multiobjective Simulated Annealing*, IEEE Transactions on evolutionary computation, **12(3)**, pp. 323–342.

[116] Smith M, 1992, *CROPWAT: A computer program for irrigation planning and management*, Irrigation and Drainager Paper No. 45, FAO, Rome, Italy.

[117] Spraque RH, 1980, *A framework for the development of decision support system*, Management Information System Quarterly, **4(4)**, pp. 1–26.

[118] Steduto P, Hsiao TC, Fereres E & Raes D, 2012, *Crop yield response to water*, Irrigation and Drainager Paper No. 66, FAO, Rome, Italy.

[119] Stegman EC, Musick JT, Hanks RJ & Watts DG, 1980, *Irrigation water management–adequate or limited water. In: Challenges of the 80's*, Proceedings of the 2th ASAE National Irrigation Symposium, October, Nebraska.

[120] Stewart JI & Hagan RM, 1973, *Functions to predict effects of crop water deficits*, Irrigation and Drainage Division, **ASCE 99(IR4)**, pp. 421–439.

[121] Stewart JI, Hagan RM, Pruitt WO, Danielson RE, Franklin WT, Hanks RJ, Riley JP & Jackson EB, 1977, *Optimizing crop production through control of water and salinity levels in the soil*, Utah Water Research Laboratory, **PRW 6-161-1**.

[122] Stewart TJ, 2007, *The essential multiobjectivity of linear programming*, ORiON, **23(1)**, pp. 1–15.

[123] Strydom MG, 1998, *Kwantifisering van Insetparameters vir die Besproeiingskeduleringsprogram BEWAB vir Aartappels en Erte*, PhD thesis, University of Free State, Bloemfontein.

[124] Sun YH, Yang S, Yeh WWG & Louie PWF, 1996, *Modeling reservoir evaporation losses by generalized networks*, Water Resources Planning and Management, **122(3)**, pp. 222–226.

[125] Tangren J, 2018, *A Field Guide to Experimental Designs*, [Online], [Cited April 2002], Available from `http://www.tfrec.wsu.edu/ANOVA/index.html`.

[126] Tanner CB & Sinclair TR, 1983, *Efficient Water Use in Crop Production: Research or Re-Search?*, pp. 1–25 in Taylor MT (Ed), *Limitations to Efficient Water Use in Crop Production*, American Society of Agronomy, Crop Science Society of America, Madison.

[127] TheGardenHelper, 2018, *The Garden Helper*, [Online], [Cited June 2012], Available from `http://www.thegardenhelper.com/planting.html`.

[128] ThoughtCo, 2018, *What is the function of plant stomata?*, [Online], [Cited June 2018], Available from `https://www.thoughtco.com/plant-stomata-function-4126012`.

[129] Toulmin C, 1985, *Livestock losses and post-drought rehabilitation in sub-Saharan Africa*, (Unpublished) Technical Report 9, International Livestock Centre for Africa (ILCA), Addis Adaba (Ethiopia).

[130] Triki E, Collette Y & Siarry P, 2005, *A theoretical study on the behavior of simulated annealing leading to a new cooling schedule*, Elsevier, **166(1)**, pp. 77–92.

[131] Troskie CG, 2012, *The economic impact of electricity price increases on the potato industry in South Africa*, PhD thesis, University of Pretoria.

[132] Umbarkar AJ & Sheth PD, 2015, *Crossover operators in genetic algorithms: A review*, ICTACT Journal on Soft Computing, **6(1)**, pp. 1083–1092.

[133] United Nations Environment Pragramme Finance Initiative, 2009, *Water-related materiality briefings for financial institutions: Agribusiness, Issue 1*, Chief Liquidity Series.

[134] van der Walt JC, 2015, *Decision support for the selection of water release strategies at open-air irrigation reservoirs*, 4^{th} year project, Stellenbosch University, Stellenbosch.

[135] van der Walt JC & van Vuuren JH, 2018, *Decision support for open-air irrigation reservoir control*, ORiON, **34(1)**, pp. 1–30.

[136] Van Vuuren JH, 2018, Professor and lecturer at *Stellenbosch University*, [Personal Communication], Contactable at `vuuren@sun.ac.za`.

[137] Van Vuuren JH & Grundlingh WR, 2001, *An active decision support system for optimality in open air reservoir release strategies*, International Transactions in Operational Research, **8(4)**, pp. 439–464.

[138] Vigeh A, 2011, *Investigation of a Simulated Annealing Cooling Schedule Used to Optimize the Estimation of the Fiber Diameter Distribution in a Peripheral Nerve Trunk*, Masters of Science thesis, California Polytechnic State University, San Luis Obispo (CA).

[139] Villiers APD, 2014, *Edge criticality in secure graph domination*, PhD Thesis, Stellenbosch University, Stellenbosch.

[140] Wardlaw R & Barnes J, 1999, *Optimal allocation of irrigation water supplies in real time*, Irrigation and Drainage Engineering, **125(6)**, pp. 345–354.

[141] Water Corporation, 2018, *What is a scheme?*, [Online], [Cited June 2018], Available from `https://www.watercorporation.com.au/home/faqs/builders-and-developers/what-is-a-scheme`.

[142] Weather SA, 2018, *South African Weather Service*, [Online], [Cited May 2018], Available from `http://www.weathersa.co.za/`.

[143] Willmott CJ, 1982, *Some comments on the evaluation of model performance*, Bulletin of the American Meteorological Society, **63(11)**, pp. 1309–1313.

[144] Winston WL & Goldberg JB, 2004, *Operations Research: Applications and Algorithms*, 4^{th} Edition, Duxbury Press.

[145] Winter TC, 1981, *Uncertainties in estimating the water balance of lakes*, Journal of the American Water Resources Association (JAWRA), **17(1)**, pp. 82–115.

[146] Wolski P, 2018, *How severe is Cape Town's drought?*, [Online; posted 23-January-2018], URL: `https://www.news24.com/SouthAfrica/News/how-severe-is-cape-towns-drought-a-detailed-look-at-the-data-20180123`.

[147] World Business Council for Sustainable Development, 2017, *Water for Business: A Guideline*, [Online], [Cited August 2012], Available from `http://wbcsdservers.org/wbcsdpublications/cd_files/datas/business-solutions/water_leadership/pdf/WaterForBusiness_Third%20Update.pdf`.

[148] Zitzler E, 1999, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Gemany: Shaker Verlag.